FEATURE RECOGNITION, OPERATIONS SEQUENCING AND NC CODE GENERATION FOR PRISMATIC PARTS WITH LINEAR SWEEP

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

ASHOK KUMAR KHUNTIA

to the

INDUSTRIAL AND MANAGEMENT ENGINEERING PROGRAMME
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

FEBRUARY, 1990

C E R T I F I C A T E

It is certified that the work contained in the thesis entitled "Feature Recognition, Operations Sequencing & NC Code Generation for Prismatic Parts with Linear Sweep" by Ashok Kumar Khuntia has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

(Dr. S. Sadagopan)
Professor
Industrial and Management Engg.
Indian Institute of Technology
Kanpur 208 016

(Dr. Kripa Shanker)
Professor
Industrial and Management Engg.
Indian Institute of Technology
Kanpur 208 016

February, 1990

ACKNOWLEDGEMENT

I express my heartfelt gratitude and sincere thanks to my thesis supervisors, Dr. Kripa Shanker and Dr. S. Sadagopan, for their constant guidance, timely help and valuable criticisms.

I would like to thank my friend Ashok Nayak with whom I had many useful discussions throughout the tenure of my project work.

I also wish to thank my friends Vishnu, Vivek, Sunil, Srinivas and Venu who have helped me both in designing and implementing the system.

Finally, my thanks are due to Mr. J. K. Mishra for excellent formatting and editing.

(Ashok Kumar Khuntia)

February, 1990

ABSTRACT

In the present work, a system has been developed for prismatic parts with linear sweep to recognize features, sequence the operations, generate NC codes and validate the system through animation. The prismatic part is considered to have orthogonal surfaces. It can have through holes to be drilled on top and bottom surfaces. The part can be machined by four types of cutters: end mill, slot, T_slot cutter and drill.

The input to the system is the two dimensional plan. The feature recognition system recognizes slots and T_slots through the comparision of X and Y-coordinates of consecutive points. Steps are identified by a sequencing algorithm. Steps may combine together to form virtual slots. The sequencing algorithm results in a sequence of operations, starting from the right end of the part and finishing at the left end of the part. The NC codes, generated by the system are verified by animating the machining of the part on the computer monitor.

For the representation of lines that constitute steps, slots and T_slots and circles that indicate holes, Array data structure is used.

The drawing is drafted in AUTOCAD ver. 10.0. The system is implemented using AUTOLISP and TURBO C ver. 2.0 on an IBM compatible PC-AT having EGA monitor.

CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 COMPUTERS IN MANUFACTURING

Manufacturing today ranks among the principal real wealth producing activities of most industrialized countries. Its primary goal is to transform the naturally available raw materials into finished products. Over the ages the manufacturing process has evolved from primitive process to very sophisticated process involving a variety of materials, tools, machines and control techniques. In the recent years the phenomenon of automation has considerably changed the nature of manufacturing activities. In the manual manufacturing activities, men supplied the power required directly and acted as a controller as well, with the tools providing mechanical aids to multiply his efforts. In semiautomatic systems which evolved later, the major portion of the power is supplied by the machines and the man senses information, interprets it and controls the machines by starting, stopping and adjusting them. In the fully automated systems which is a recent phenomenon, both the power supply and control functions are performed by the machines. The man's role is merely that of a monitor to help control the processes and maintain surveillance over the crucial parameters.

Computers have played a very important role in automation. The first impact of computers in manufacturing was felt in the early fifties with the introduction of numerically controlled (NC) machines. Numerical control is method of automatic control that uses symbolically coded instructions to cause a machine to perform a specific series of operations. The logical extension of NC was the Computerized Numerical Control (CNC) where a general purpose mini computer replaced the the specialized control unit. In the seventies, manufacturing activity gained significantly by the introduction of industrial robots. Such developments led to a new system of manufacture generally known as Computer Aided Manufacturing (CAM).

Computers have played an equally important role in automating the design activity as well. Computer Aided Design (CAD) systems support the design process at all levels conceptual, preliminary and final design. A geometrical model can be used to visualize the part while the features of the products such as strength, stiffness etc. can be tested against several environmental conditions using computer algorithms and data structures.

While Computer Aided Design (CAD) is concerned exclusively with the design activity; Computer Aided Manufacturing (CAM) is concerned with the machines and equipment that are directly involved in the manufacture of the product. It is the process planning function that translates the design data into work instructions to produce the part. Computers are playing an increasing important role in the recent years in this activity as well. Computer Aided Process Planning (CAPP) as it is generally

called, integrates the Computer Aided Design (CAD) output with the Computer Aided Manufacturing (CAM) equipment. This holistic scenario viz. Computer Aided Design (CAD), Computer Aided Process Planning (CAPP) and Computer Aided Manufacturing (CAM) would form the series for Computer Integrated manufacturing Systems (CIMS)-an emerging view of contemporary manufacturing.

1.2 PROCESS PLANNING

Process planning is that function within manufacturing facility that establishes the machine process and parameters that are to be used to convert a part from its initial raw material form to a final form predetermined by the design function. It primarily involves

(a) Identification of the actual machining process

(b) Selection of machine tools

(c) Selection of tools, jigs, fixtures etc.

(d) Determination of optimal process parameters (for example feed rate, speed, depth of cut in the case of metal cutting)

(e) Selection of an optimal sequence of operations.

The above steps of process planning require different strategies depending upon whether the part geometry is axi-symmetric (rotational) or axi-nonsymmetric (prismatic). An axi-symmetric part is generated by rotating a set of lines and arcs about an axis whereas a prismatic part is generated by sweeping a polygon consisting of lines and arcs about any of the X, Y, and Z axis (linear sweep) generating flat surfaces, or along all the axes

simultaneously (nonlinear sweep) generating nonlinear curved surfaces. The axi-symmetric parts have mostly the concentric circles as their end views whereas the prismatic parts contain complex features such as slots, T-slots, steps, grooves, dovetails etc. The axi-symmetric parts are usually produced on the lathe type of machine tools and prismatic parts require machine tools having several axes of motion and associated controls. The steps in process planning for prismatic parts, in general, appear to be more cumbersome than those for axi-symmetric parts.

Traditionally, process planning has been done manually which requires considerable amount of efforts on the part of specially skilled and vastly experienced shop floor personnel. The advent of computers and the emerging database technology have proved to be of great help in this computer decision making process planning activity. The resulting body of knowledge is generally known as Computer Aided Process Planning (CAPP). The two major approaches to process planning are (Chang & Wysk, [1]) : Variant Approach and Generative Approach. These are briefly described in the following paragraphs.

VARIANT PROCESS PLANNING

The variant approach to process planning is a computer assisted extension of the manual approach. In this method the process plan for a new part is generated by recalling, identifying and retrieving an existing plan for a similar part and making the necessary modifications for the new part.

The variant approach has several advantages. The data management and retrieval of the computer greatly reduces the time

consumption and tedium of paper handling and hand copying work. Secondly the need to code parts into families helps insure that the standardized process plans will be more consistent. The disadvantages of this approach are: (i) when new machines or processes become available, the existing database must be reviewed and revised manually by the process planner, (ii) the system cost and the time required to classify, code and store the data are enormous, (iii) an experienced process planner is a must to update and modify the database and process plans.

GENERATIVE PROCESS PLANNING

The generative approach to process planning utilizes an automated computerized system consisting of decision logic and geometry based data to uniquely determine the processing decisions for converting a part from a rough to a finished state.

A generative process planning system essentially consists of two major components. The first is a geometry based coding scheme for translating physical features and engineering specifications into computer interpretable data. The second component comprises of decision logic to compare the part geometry requirements with manufacturing capabilities and availabilities.

The generative method of process planning has many advantages in terms of automating the manual activities such as in creating manufacturing cost estimates, routings and instructions, thereby producing results which are more accurate, consistent and inexpensive. Its most important distinguishing feature is that it is fully automatic and that an up-to-date operation sheet is

generated each time a part is ordered. However this method has also its drawbacks. Designing and developing a generative process planning is a formidable task. It requires long term investment of time and manpower. In spite of all these difficulties, the development of a generative process planning system should be viewed in the light of the future needs for integrated design and manufacturing functions in the environment characterized by frequently changing part design, and development of need, methods and machines.

The present work is concerned with generative approach of process planning.

1.3 REVIEW OF PREVIOUS WORK

There have been considerable research and developmental efforts in the field of generative process planning that can be attributed to have brought about the integration of CAD and CAM. A brief survey of some of the salient developments are presented here.

In 1977, Wysk developed a stand-alone generative system known as APPAS (Automated Process Planning and Selection) at Purdue University for planning milling and hole cutting operations on machining centers. This work was later extended by Chang, Wysk and Davis by linking APPAS with an interactive computer graphics terminal to demonstrate the concept of an integrated CAD and process planning system.

A number of sophisticated planning systems are currently being used in industry to plan turning operations on cylindrical parts. The most technologically advanced system in use today is probably

the Computer Managed Process Planning (CMPP) system for cylindrical workpieces developed by United Technologies Corporation in conjunction with the U.S. Army Missile Command. CMPP has been cited as a major breakthrough in the marriage of CAD and CAM. CMPP automates the link between CAD and CAM by being able to accept geometric part data from a CAD system and perform planning functions to generate manufacturing process documentation, drawings, or NC programming data.

UNIGRAPHICS II developed by McDonnel Douglas Information Systems of U.S. integrates CAD & CAM to a limited extent. The user has to specify the features and machining parameters. The NC codes are generated automatically.

1.4 SCOPE OF THE PRESENT WORK

The present work deals with prismatic parts with linear sweep having through holes in it. It aims at extracting features from a two dimensional plan view of the engineering drawing of the part, selecting tools for machining different features, sequencing the operations involved and generating NC codes for those operations. The correctness of the NC codes is verified by animating the machining of the part. Although the restriction to the orthogonal surfaces of prismatic parts appears to be a constraint, still it serves to illustrate the basic issues involved. And, it is not very uncommon to find such prismatic parts in industry.

1.5 ORGANIZATION OF THE THESIS

Chapter II discusses about drafting a drawing and retrieving the drawing information in the form a file.

Chapter III discusses about the way the points of the part drawing are arranged in a particular order and how that order helps in identifying features such as slots and T_slots.

Chapter IV discusses the methodology of dividing the part drawing into four sides and the way, the steps are identified and the sequencing of the various operations is done.

Chapter V discusses about selection of tools, generation of NC codes, collection of statistics for the machining operations and verification of NC codes.

Chapter VI explains the different steps of the system through an illustration.

Chapter VII narrates the scope for future work and gives the conclusive remarks.

Chapter VIII contains the users' manual.

CHAPTER II

DRAWING DRAFTING AND FILE RETRIEVAL

This chapter presents the methodologies for part drawing drafting and retrieval and decoding of the file containing the information about the part drawing.

2.1 DRAFTING THE DRAWING

Part design characteristics such as shape, size, dimensions, surface finish, tolerance etc. need to be represented in a manner convenient and suitable for subsequent analysis for manufacturing. This representation including the major dimensional characteristics viz. size and shape is referred to as drafting in the present context.

A part drawing can be drafted in several ways. A prismatic part, where the sweep is along the Z-direction and plan is a polygon on X-Y plane, can be be drafted : (i) by joining points in a sequential manner; (ii) by drawing lines in a sequential manner; (iii) by drawing a set of primitives like step, slot and T_slots in a sequential manner; (iv) by drawing lines in an arbitrary manner and establishing their logical sequence to form the polygon; or (v) by simply copying the whole polygon from a previously drafted drawing. In the first three methods the user is constrained to draw the lines in a particular sequence and any mistake during drafting necessitates redrawing from the beginning. However, in the fourth

method, lines can be drawn in any manner found convenient for example, all vertical lines may be drawn one after the other followed by horizontal lines, and vice versa. Further, in the event of any mistake, a line can be deleted and redrawn at any time during drafting without redrafting the whole drawing. In the present work, this method is chosen for further discussion, and implementation. A drawing drafted using the fifth method is also acceptable in the present implementation. Similarly, circles can be drafted individually or can be copied from a previously drafted drawing.

Most of the general purpose computer programming languages can be used for drafting. To serve the specialized needs of drafting, several packages have been developed. One of the most popular package is AUTOCAD. In the present work, AUTOCAD version 10.0 is used for drafting. The reasons for choosing AUTOCAD as the drafting package are as follows:

AUTOCAD runs in the DOS operating system in a PC-XT/AT class of machines which are popular and easily accessible. Besides this, AUTOCAD design package is known to be a powerful drawing tool. It follows instructions quickly and produces the exact drawing. AUTOCAD features allow the user to correct drawing errors easily and one makes revisions without redoing the entire drawing. It also offers wide flexibility in drafting of drawing. For example, a closed polygon can be drawn as a set of lines or by a single polyline.

This drafted drawing information is stored in files for the next manufacturing planning activity, that is, process planning.

2.2 CREATION OF INTERCHANGE FILE

After the drawing is drafted, it is checked (by a software) to see whether the entities (lines and circles) have got individual identities or group identities. In case an entity does not have individual identity (it would then have got some group identity), then it is processed further. For example, more than one line may have been drawn as a polyline, or a whole polygon with some circles inside it may have been drawn as a block. In such cases, for the ease of analysis, the polyline or block is exploded into individual entities (lines and circles) for creating individual identities. To create the interchange file (described in the next section), each entity in the drafted drawing must have individual identity.

2.3 FILE FORMAT

Drafting packages store the drawing information in a very compact form which varies significantly from package to package. However, these drawing information are often required to be accessed by users through programs written in some other languages and also through packages. For example, information stored in AUTOCAD can be accessed through a program written in C or in dBase III. With this in view, standard formats have been developed to store the drawing information. Some of the popular file formats (AUTOCAD Reference Manual, [2]) are:

(a) Drawing Interchange File Format (DXF),

(b) Binary Drawing Interchange Files (DXB),

2.3.1 DRAWING INTERCHANGE (DXF) FILES

DXF files are standard ASCII text files which can be easily analyzed by other programs. A DXF is composed of a multiplicity of groups, each of which occupies two lines in the DXF file. The first line of a group is a group code, which is a positive nonzero integer and the second line of the group is the group value. The type of value for a group code is derived as given in the following table.

Group code	Type of value
0 - 9	String
10 - 59	Floating point
60 - 79	Integer
210 - 239	Floating point
999	Comment (string)

A DXF file has the extension ".dxf". The overall organization of a DXF file comprises of four different sections; header section, tables section, block section and entities section; in that order. These file sections are separated by file separator groups.

(1) HEADER SECTION. This section contains settings of variables associated with the drawing. Each variable in the header section is specified by a group code of 9 giving its name followed by group(s) that supply its value. For example, the AUTOCAD version number is specified as follows:

9	<group code>
$ACADVER	<variable for AUTOCAD version number>
1	<group code> (indicates that the next line has the value of the variable appearing immediately before this line)
10.0	<version number> (value of the variable $ACADVER)

The details of other variables which are specified in this section are: text style ($TEXTSTYLE), the size of the screen ($LIMMIN, $LIMMAX), unit of measurement used ($UNIT), thickness of polyline($THICKNESS) AND color ($COLOR) of different layers etc.

(2) TABLES SECTION. This section contains the following tables specifying the definitions of the corresponding items.

(a) Linetype table	LTYPE
(b) Layer table	LAYER
(c) Textstyle table	STYLE
(d) View table	VIEW
(e) User coordinate system table	UCS
(f) Viewport configuration table	VPORT
(g) Drawing manager table	DWGMGR

Each table is introduced with a group code of 0 with the label "TABLE". This is followed by a group code of 2 identifying the particular table and a group code of 70 that specifies the maximum number of entries that may be included in the table. For example, for a drawing which has 8 linetypes, the LTYPE table is specified as

```
0              <group code>
TABLE
2              <group code>
LTYPE
70             <group code>
8              (number of entities in the table LTYPE)
```

(3) BLOCK SECTION. A block is a set of entities grouped together into a compound object. Once so grouped, the entities are given a block name. This name can be used to insert the group of entities anywhere in the drawing. The block section contains the specification of various blocks. The block specification may include name and number of blocks, number and description of entity(ies) in a block, and so on. A block is specified as follows:

```
0              <group code>
BLOCK
8              <group code>
SCRATCH        <Block name>
2              <group code>
BLK1           <Block name>
...
```

(4) ENTITIES SECTION. An entity in AUTOCAD is a predetermined element that can be drawn by means of a single command. Examples of entities are lines, arcs, circles, text, dimensions etc. The entities section contain entity description that begins with a group code of 0, identifying the entity type. Then it contains a group code of 8 that gives the name of the layer on which the entity resides. Each entity may have elevation, thickness or color

type information associated with it. The end of an entity description is indicated by the next 0 group, beginning the next entity or indicating the end of the section. For example, a line is specified as follows:

```
0           <group code>
LINE
8           <group code>
CADM$10     <Layer number>
10          <group code>
246.11      <X-Coordinate>
20          <group code>
190.78      <Y-Coordinate>
...
```

The end of file is marked with EOF.

2.3.2 BINARY DRAWING INTERCHANGE FILES

In Binary drawing interchange files data are represented in binary form. These files are of two types: Binary DXF files, and Binary DXB files.

[1] <u>Binary DXF files</u>

A binary DXF file contains all the information present in an ASCII DXF file, but in a much more compact form. Unlike ASCII DXF file, which entails a trade-off between the size (number of entities) and float-point accuracy, (limited to sixteen places after decimal point), binary DXF files preserve the accuracy in the drawing information upto the specified value. The disadvantage of this format is that it is supported only by AUTOCAD version 10.0

and it cannot be read by earlier versions.

Here, the group code is a single-byte binary value, and the value that follows is one of the following:

(1) a two-byte integer with the least significant byte first and the most significant byte last,

(2) an eight-byte IEEE double precision floating-point number stored with the least significant byte first and the most significant byte last, or

(3) an ASCII string terminated by a zero (NULL) byte.

[2] Binary DXB files

This is yet another file format more compact than the binary DXF file. However it has limitations on the number of entities it can represent. AUTOCAD has a facility to read such files, but has no direct method of writing them.

2.4 DECODING THE DXF FILE

The drawing information has to be analyzed for manufacturing. So the DXF file is decoded to retrieve the relevant drawing information about the entities such as lines and circles. The data structure used to store these information is Array. The reasons for using Array as the data structure is discussed in the next paragraph.

Arrays and linked lists are generally used to store a number of records of the same type. Comparing between linked list and array, the representation of an array is compact because it takes less space in the computer memory than the linked list to store the same amount of data. Besides this, searching is very fast in a sorted array compared to a linked list.

The different records and arrays defined to store the information about lines, circles, and circular arcs are:

RECORDS

RECORD	FIELD(S)	TYPE	DESCRIPTION
pt_info	x	real	X-coordinate
	y	real	Y-coordinate
line	stpt	pt_info	Start point
	endpt	pt_info	End point
circle	center	pt_info	center
	radius	real	radius
arc	center	pt_info	center
	radius	real	radius
	stangle	real	start angle
	endangle	real	end angle

ARRAYS

ARRAY NAME	RECORD TYPE
lines	line
arcs	arc
circles	circle

The three arrays lines, arcs and circles contain the information about all the lines, arcs and circles of the part drawing.

CHAPTER III

FEATURE EXTRACTION & RECOGNITION: SLOTS AND T_SLOTS

Feature extraction constitutes an important and at the same time complex activity of CAD/CAM integration. In the present context of prismatic parts with a linear sweep the features considered are steps, slots and T_slots. These features are to be extracted from a two-dimensional drawing. In this chapter, the methodology related to identification of slots and T_slots are discussed. The identification of steps and other related issues are discussed in the next chapter. A brief review of of the previous work on extraction of features in general, is presented below.

As mentioned by Apte [3], the following are some of the major contributions in feature extraction of general features.

Kakino	Part description method in which the machined components are represented as a linear concatenation of adjacent elements. The elements consist of geometric primitives and operators (translation and revolution).
Woo	A technique known as alternative sum of volumes (ASV) which extracts features from the boundary representation of a 2-D or 3-D parts.

Kyprianou	A solid modeller boundary representation database to extract information regarding the features. Devised a feature language which can describe all possible shapes. This technique is known as syntactic pattern recognition.
Choi	Syntactic pattern recognition in identifying elementary machined surfaces. A 3-D workpiece can be decomposed into machine surfaces or features which can be described as 2-D entities.
Henderson	An expert system which recognizes the cavity features and extracts those features as a solid volume of material to be removed.

3.1 THE APPROACH

The methodology of identification of slots and T_slots as proposed to be implemented in the present work has been described in this section. Circles represent holes inside the part and thus do not constitute a part of the boundary of the part drawing. So only lines corresponding to the edges of the polygon representing plan of the part on X-Y plane, are considered for the identification of slots and T_slots. The steps described in the next paragraph are common to the identification of both slots and T_slots. The additional steps for slots and T_slots will be discussed individually.

As mentioned in section 2.1, the lines to draft the polygon are drawn in a random manner. Slots and T_slots consist of a set of consecutive lines. Thus for the identification of all the slots and

T_slots of the part drawing the lines need to be first ordered in a sequential manner.

A line is represented by its two end points and a polygon by an ordered set of lines. However, a polygon can also be considered as a set of points. In keeping the information of a closed polygon in terms of its lines the data about a point appears twice. Thus, in order to keep the data storage to a minimum, the polygon is represented in terms of its vertices.

The data structure to store vertices of the polygon is an array of records as described in the next paragraph.

RECORD	FIELD	TYPE	DESCRIPTION
modpt_info	ptno	integer	point index
	x	real	X-coordinate
	y	real	Y-coordinate
	side	integer	side index

The array to store information about all the vertices of the polygon is defined as:

seqpoints array of record type modpt_info

The procedure to obtain the array of ordered vertex points of the polygon from the input of the lines given in random manner is described below.

procedure: sequence_points

Input:

lines, an array of line (line is represented by two end points)

Output:

seqpoints, an array of points, either in a clock wise or in an anti-clock wise direction.

Algorithm:

Step 1

Start with any extreme point (P) of a line. Designate the point to be the first record of the array seqpoints.

Step 2

For each record in the array lines do

Step 2(a)

Find the line for which P is either a start point or an end point.

Step 2(b)

If P is a start point in the line found, then

update the coordinates of P to the coordinates of the end point of the line found.

else if P is an end point in the line found, then

update the coordinates of P to the coordinates of the start point of the line found.

Step 2(c)

Delete the line found from the array.

The above procedure will give an array seqpoints which will contain the information of all vertex points of the polygon. The issue of orientation of points in clockwise or in an anti-clockwise direction will be discussed later.

3.2 RECOGNIZING SLOTS

A slot can appear on any of the four sides of the part drawing parallel to the Z-axis. For reference in the subsequent sections, as shown in Figure 3.1, the left hand side of the part drawing is referred to as side 1, the top side as side 2, the right hand side as side 3 and the bottom side as side 4. As can be seen, four points are required to identify a slot.

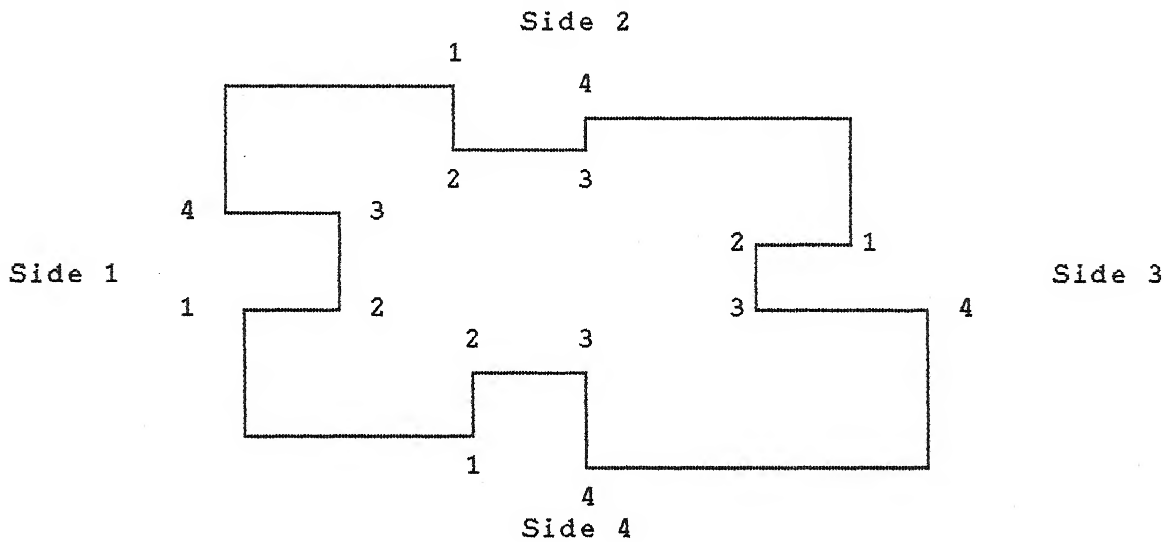


Figure 3.1 : Representation of Slots

In order to represent a slot, pt[i] is defined as a record of type pt_info where i is an integer. In order to identify slots on sides 1 and 3, all or some of the following three conditions are necessary and a suitable combination of these along with some additional conditions used identify the slots on these sides.

```
COND_1 :    pt[1].y = pt[2].y;
            pt[2].x = pt[3].x;
            pt[3].y = pt[4].y;
COND_2 :    pt[1].x > pt[2].x;
            pt[4].x > pt[3].x;
```

```
COND_3 :    pt[1].x < pt[2].x;
            pt[4].x < pt[3].x;
```

A slot on sides 1 and 3 has to satisfy COND_1 and either COND_2 or COND_3. Similarly, in order to identify slots on sides 2 and 4, altogether the following three conditions are necessary and a suitable combination of these along with some additional conditions used identify the slots on these sides.

```
COND_4 :    pt[1].x = pt[2].x;
            pt[2].y = pt[3].y;
            pt[3].x = pt[4].x;
COND_5 :    pt[1].y > pt[2].y;
            pt[4].y > pt[3].y;
COND_6 :    pt[1].y < pt[2].y;
            pt[4].y < pt[3].y;
```

A slot on side 2 and side 4 has to satisfy COND_4 and either COND_5 or COND_6. However, these conditions are necessary but not sufficient to identify a slot. This will be clear from Figure 3.2 which depicts the difference between slots and projections.

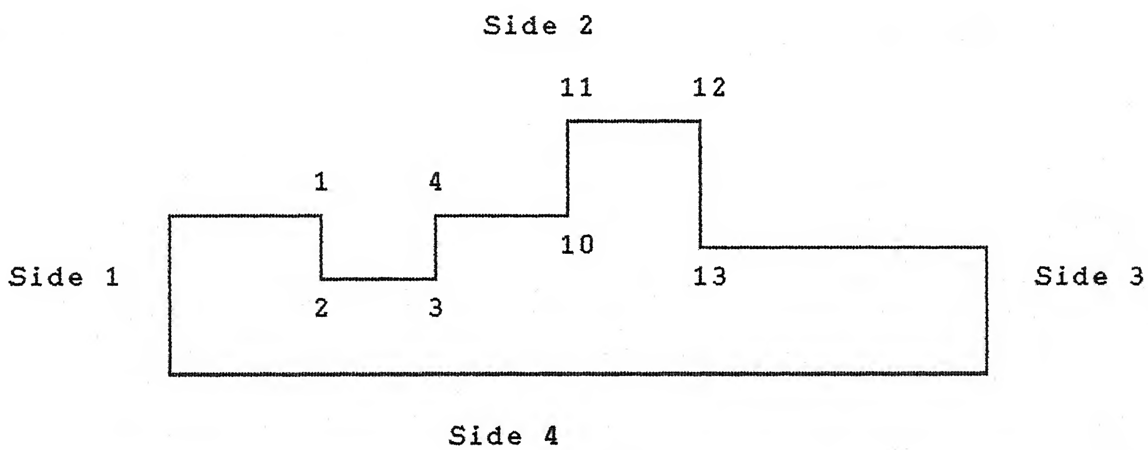


Figure 3.2 : Difference between Slot and Projection

Points 1, 2, 3, 4 satisfy COND_4 and COND_5, so 1-2-3-4 forms a slot. Points 10, 11, 12, 13 satisfy COND_4 and COND_6 and hence 10-11-12-13 is also identified as a slot. However points 10, 11, 12, 13 do not form a slot (rather they form a projection). The same problem arises on the other sides also. So a slot has to be distinguished from a projection.

It is observed that if the points are sequenced in a clockwise manner then any point inside the part drawing always lies to the right side of the direction of orientation, and if the points are sequenced in an anti-clockwise manner then a point inside the part drawing always lies to the left. Mathematically, if the points are sequenced clockwise, and a set of four consecutive points satisfy the required conditions mentioned above and if the direction vector corresponding to the area enclosed by the points is positive, then these points constitute a slot. A negative direction vector, on the other hand, identifies a projection. So there is a need to sequence the points either in a clockwise manner or in an anti-clockwise manner.

It is observed that when the direction vector is negative for the area corresponding to the points: the point previous to the leftmost point, the leftmost point (i.e. having minimum X coordinate), and the point next to the leftmost point the points are sequenced in a clock wise manner. And, the direction vector is positive if the points are sequenced in an anti-clockwise manner. The steps to sequence the points either in a clockwise or in an anti-clockwise manner are as described in the next paragraph:

Step 1

The index of the point having minimum X-coordinate is found. Let this point be called leftmost_pt.

Step 2

The points are already there in a particular sequence, but it is not known whether they are in a clockwise manner or in an anti-clockwise manner. Whatever it may be the point previous to leftmost_pt (called prev_pt) and the point next to leftmost_pt (called next_pt) are identified.

Step 3

The direction vector is given by the vector product of the two lines joining pairs of points the prev_pt and leftmost_pt, and leftmost_pt and next_pt. Let the three points be indexed as 1, 2 and 3. The vector product of lines joining 1-2 and 2-3 is given by

```
vectorprod(pt[1],pt[2],pt[3]) =
        (pt[2].x - pt[1].x) * (pt[3].y - pt[2].y)
      - (pt[3].x -pt[2].x) * (pt[2].y - pt[1].y)
```

The above vector will be along Z-direction. If the magnitude is positive then the vector is along the positive Z-direction and if the magnitude is negative then the vector is along the negative Z-direction.

Step 4

If the vector product is negative (i.e. the points are sequenced anti-clockwise) then the order of the points is reversed and if the vector product is positive then the

points are already sequenced in a clockwise manner.

The above steps are summarized in the following procedure.

procedure Recognize_slots

Input:

seqpoints, array of points

Output:

array of slots where a slot is an array of four point indexes.

<u>Algorithm:</u>

Step 1

Let i be initialized to 0.

Step 2

For all points do

```
if (   ( COND_1 AND (COND_2 OR COND_3) OR
       ( COND_4 AND (COND_5 OR COND_6) ) AND
       ( vector_prod(seqpoints[i], seqpoints[i+1],
                       seqpoints[i+2] ) > 0 )
then  a slot is identified by point indexes i, i+1,  i+2  and
      i+3.
```

Step 3

Increment i by 1.

3.3 RECOGNIZING T_SLOTS

Like a slot, a T_slot can appear on any of the four sides of the part drawing.

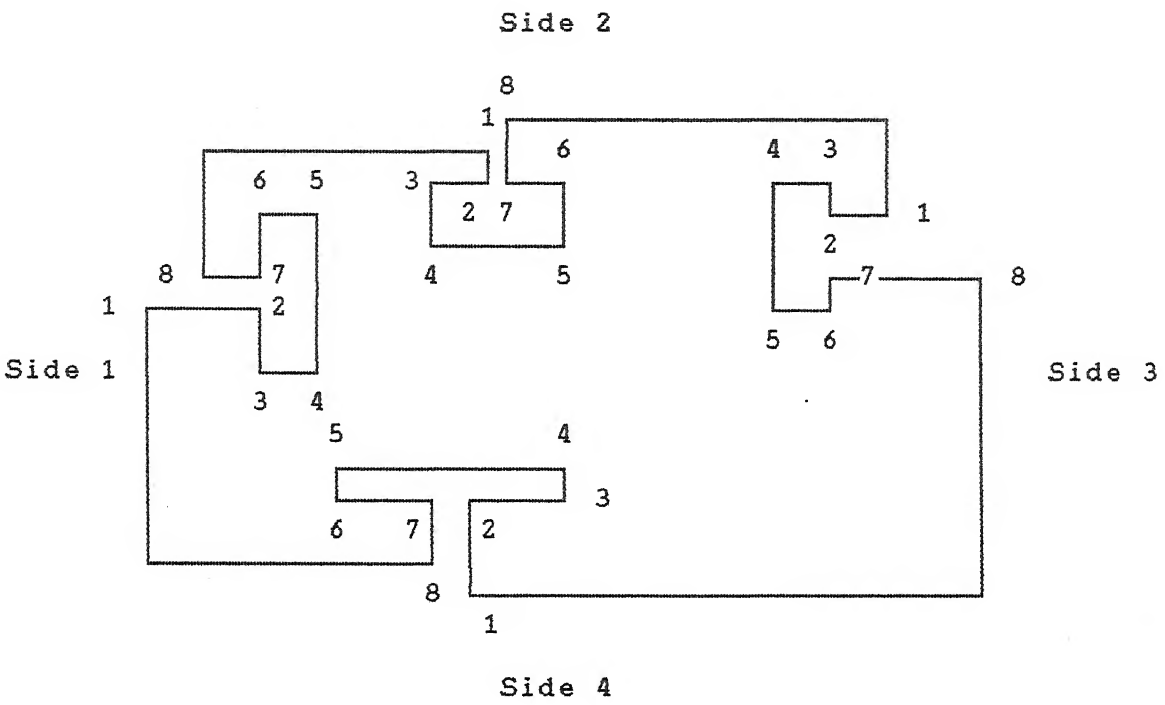


Figure 3.3 : Representation of T_slots

As shown in Figure 3.3, a T_slot is identified by 8 points. A T_slot on side 2 and side 4 has to satisfy the following conditions.

```
COND_1 :  pt[1].x = pt[2].x;
          pt[2].y = pt[3].y;
          pt[3].x = pt[4].x;
          pt[4].y = pt[5].y;
          pt[5].x = pt[6].x;
          pt[6].y = pt[7].y;
          pt[7].x = pt[8].x;
COND_2 :  pt[8].x - pt[1].x = pt[7].x - pt[2].x;
COND_3 :  pt[2].x - pt[3].x = pt[6].x - pt[7].x;
COND_4 :  abs(pt[5].x - pt[4].x) > abs( pt[8].x - pt[1].x);
COND_5 :  pt[3].y - pt[4].y = pt[6].y - pt[5].y;
```

For side 2 and side 4 the above condition i.e. from COND_1 to COND_5 are necessary and sufficient to identify a T_slot. To show that the above conditions are necessary for a T_slot, we consider the combinations of a subset of conditions as shown in Figure 3.4.

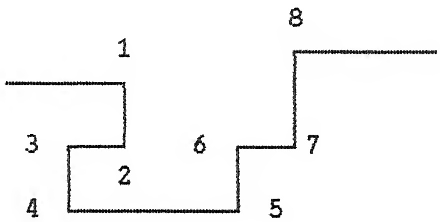


Figure 3.4(a)

This feature will be identified as a T_slot without COND_3.

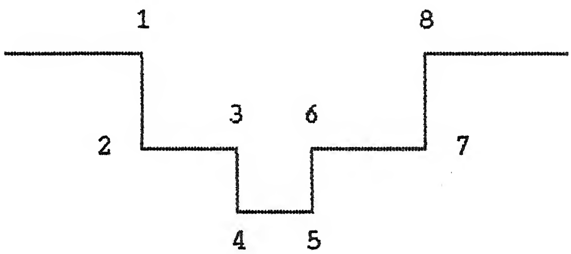


Figure 3.4(b)

This feature will be identified as a T_slot without COND_4.

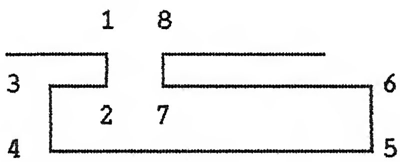


Figure 3.4(c)

This will be identified as a T_slot without COND_5.

Figure 3.4 : T_slot Identification and Necessary Conditions

Similarly, for sides 1 and 3, the different conditions to identify a T_slot are as follows:

```
COND_1 :  pt[1].y = pt[2].y;
          pt[2].x = pt[3].x;
          pt[3].y = pt[4].y;
          pt[4].x = pt[5].x;
          pt[5].y = pt[6].y;
          pt[6].x = pt[7].x;
          pt[7].y = pt[8].y;
COND_2 :  pt[8].y - pt[1].y = pt[7].y - pt[2].y;
COND_3 :  pt[2].y - pt[3].y = pt[6].y - pt[7].y;
COND_4 :  abs(pt[5].y - pt[4].y) > abs( pt[8].y - pt[1].y);
COND_5 :  pt[3].x - pt[4].x = pt[6].x - pt[5].x;
```

It may be noted that according to the procedures developed, all the T_slots will be identified as slots also. In Figure 3.3, the points 3, 4, 5 and 6 of a T_slot is also identified as a slot. To identify T_slots and slots exclusively and separately, T_slots that also appear in the list of slots are to be deleted from the list of slots.

The present methodology appears to be simpler than the one adopted by Apte [3] which uses syntactic pattern recognition approach. However, In Aptes's approach, the surfaces are not restricted to be orthogonal.

CHAPTER IV

SEQUENCING

Different machining operations are involved in converting the raw stock into the desired shape and size. These machining operations should be sequenced in a manner such that the sequence is not only feasible but also optimum. In the present work, as mentioned earlier, the part to be manufactured is prismatic with linear sweep. It is assumed that the part has to be manufactured out of a raw stock (blank) whose plan view is a rectangle on X-Y plane and height (distance along Z-direction) is same (or more) as that of the part. It is also assumed that all the machining operations are to be done in a vertical milling machine. The first machining operation would be to remove all the extra material along X,Y and Z-directions so as to obtain a rectangle just enveloping the part. The next machining operation would be to drill the holes if they exist. Then features like step, slot and T_slot are to be machined. A slot or T_slot can face to the right, left, top or bottom. But, in a vertical milling machine a slot or T_slot can be machined only when it faces to the top. So the part has to be fixtured in four different orientations so that all the features can be machined. This requirement gives rise to the necessity of dividing all the points onto four different sides.

4.1 DIVIDING THE PART DRAWING INTO FOUR SIDES

As mentioned earlier and also in the previous chapter, slots and T_slots can be machined only from the top. Slots and T_slots facing to the left will be in side 1, those facing top will be in side 2,those facing to the right will be in side 3 and those facing down will be in side 4. The following procedure (PROC1) divides the features over four different sides.

PROC1

procedure: put_points_in_different_sides

Input:

seqpoints, an array of points (Each point is represented by the X,Y coordinates and point index)

Output:

seqpoints, an array of points (Each point is represented by the X,Y coordinates,point index and side index)

(The array seqpoints, is used for input as well as output. When used for input the value of side index is initialized as -1)

Algorithm

Step 1

For each slot do

If slot faces to the left

then update the side index of all four points of the slot to 1.

else if slot faces top

then update the side index of all four points of the slot to 2.

else if slot faces to the right

then update the side index of all four points of the slot to 3.

else if slot faces down

then update the side index of all four points of the slot to 4.

Step 2

For each T_slot do

If T_slot faces to the left

then update the side index of all eight points of the T_slot to 1.

else if T_slot faces top

then update the side index of all eight points of the T_slot to 2.

else if T_slot faces to the right

then update the side index of all eight points of the T_slot to 3.

else if T_slot faces down

then update the side index of all eight points of the T_slot to 4.

The above procedure identifies the points corresponding to slots and T_slots over four sides. The points not entertained by the above procedure have to be assigned their respective sides. The following procedure is suggested for such left-over points.

The rectangle just enveloping the plan of the part drawing is divided into four portions (A,B,C,D), as shown in Figure 4.1. The left-over points in portions A, B, C, D are allocated sides 1, 2,

3, and 4 respectively. That a point lies in a particular portion is decided as follows.

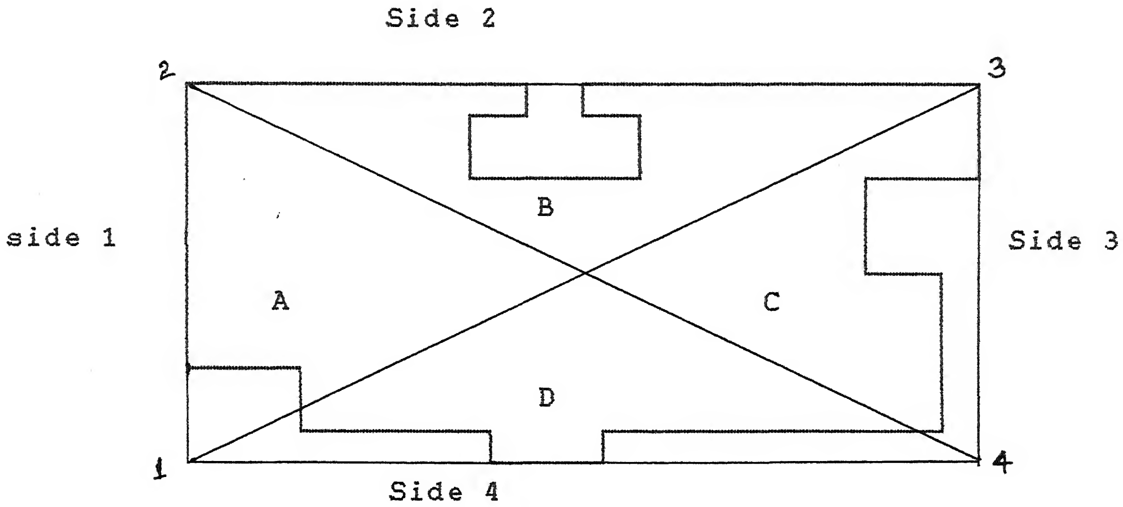


FIGURE 4.1 : Allocating Side Index for Left-over Points

The two diagonals of the rectangle just enveloping the part drawing are identified (they are 1-3 and 2-4 here). The decision table for a certain point to be in a particular side is as:

Left of 1-3 Left of 4-2	Left of 1-3 Right of 4-2	Right of 1-3 Left of 4-2	Right of 1-3 Right of 4-2
side 1	side 2	side 3	side 4

The concept of vector product, used earlier is applied here also to determine as to which side of a diagonal a particular point lies. Considering the diagonal 1-3, the vector product of line joining point 1 and a point x and line joining point x and point 3 will be positive if the point x lies to the left of the diagonal 1-3 and it will be negative if the point x lies to the right of the diagonal 1-3. Using the above concept, the following procedure

(PROC2) puts the left out points over the four sides.

PROC2

procedure: put_left_over_points_into_diff_sides

Input:

seqpoints, an array of points (Each point is represented by the X,Y coordinates, point index and side index)

Output:

(the same) array seqpoints, after modifying the side index of some points.

ALGORITHM

Step 1

For each point (i is the point index) having side index -1 do

Step 1.1

```
if vector_prod(min_x,min_y,seqpoints[i].x,
                seqpoints[i].y,max_x,max_y)   >   0
                (for diagonal 1-3)
     left1 = TRUE;
else left1 = FALSE;
if vector_prod(max_x,min_y,seqpoints[i].x,
                seqpoints[i].y,min_x,max_y) > 0
                (for diagonal 2-4)
     left2 = TRUE;
else left2 = FALSE;
```

Step 1.2

```
if ( left1 AND left2 ) seqpoints[i].side = 1;
else if ( left1 AND (NOT left2) ) seqpoints[i].side = 2;
```

```
else if ((NOT left1) AND (NOT left2)) seqpoints[i].side = 3;
else seqpoints[i].side = 4;
```

After applying the two procedures put_points_in_different_sides and put_left_over_points_into_diff_sides, to all the points it is checked if the consecutive points (excepting the corner points) fall on the same side or not. For example,in the part drawing shown in Figure 4.2, point 3 will on side 1 and all points from 1 to 16,except point 3 will be on side 2. So in such cases point 3 is also put on side 2.

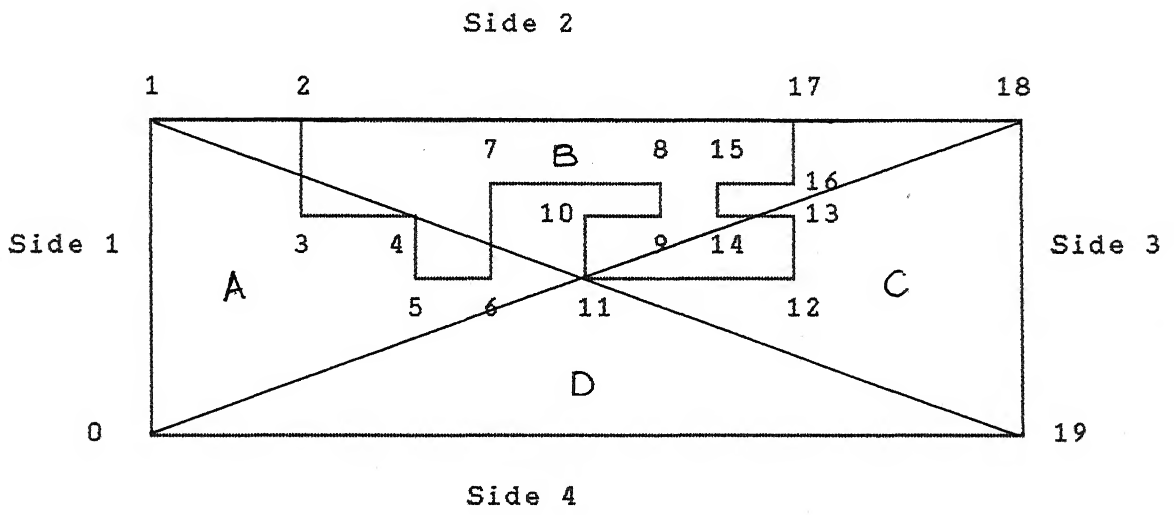


Figure 4.2 : Outlier Points of Procedures PROC1 and PROC2

After all the points are assigned a side index, points of each side are stored for subsequent uses in a different array of pt_info records. Let these four arrays be side[1], side[2], side[3] and side[4].

4.2 TRANSFORMING THE COORDINATES

On a 3-axis vertical milling machine, a side has to face upward for machining. The part is oriented in a manner such that the plan view becomes the elevation. For this, the coordinates of points change (except for side 2) when the corresponding sides face upwards. It is assumed that at any orientation of the part drawing the minimum values of x and y do not change i.e. the bottom left reference point is always retained. Figure 4.3 illustrates the concept of required rotation.

The coordinates of the points on different sides are transformed in the following way.

```
        for side 1
side[1][i].x =  min_x + seqpoints[ side[1][i].ptno ].y - min_y;
side[1][i].y =  max_x - seqpoints[ side[1][i].ptno ].x + min_y;
        for side 3
side[3][i].x =  min_x - seqpoints[ side[3][i].ptno ].y + max_y;
side[3][i].y =  min_y + seqpoints[ side[3][i].ptno ].x - min_x;
        for side 4
side[4][i].x =  min_x - seqpoints[ side[4][i].ptno ].x + max_x;
side[4][i].y =  min_y - seqpoints[ side[4][i].ptno ].y + max_y;
```

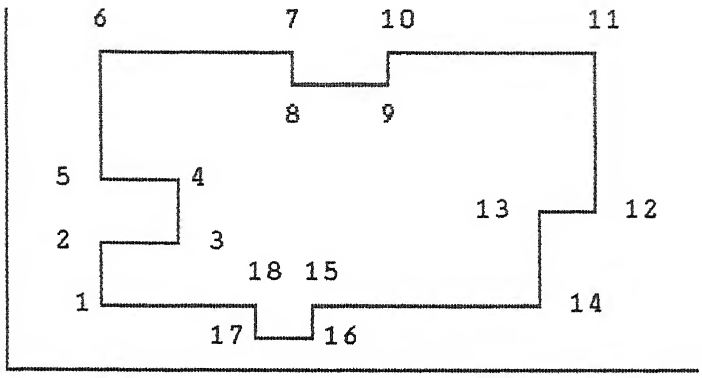


Figure 4.3 (a) : Plan

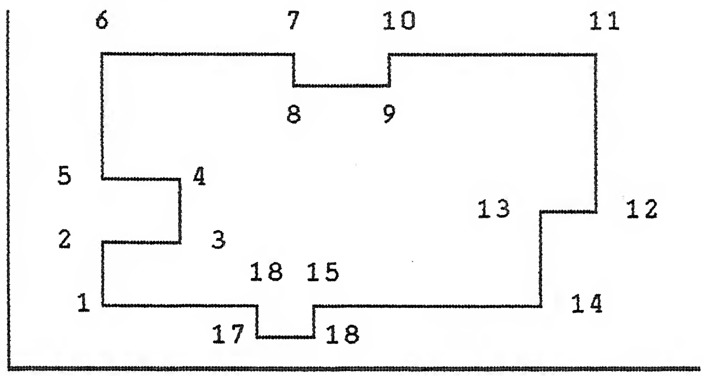


Figure 4.3 (b) : Elevation

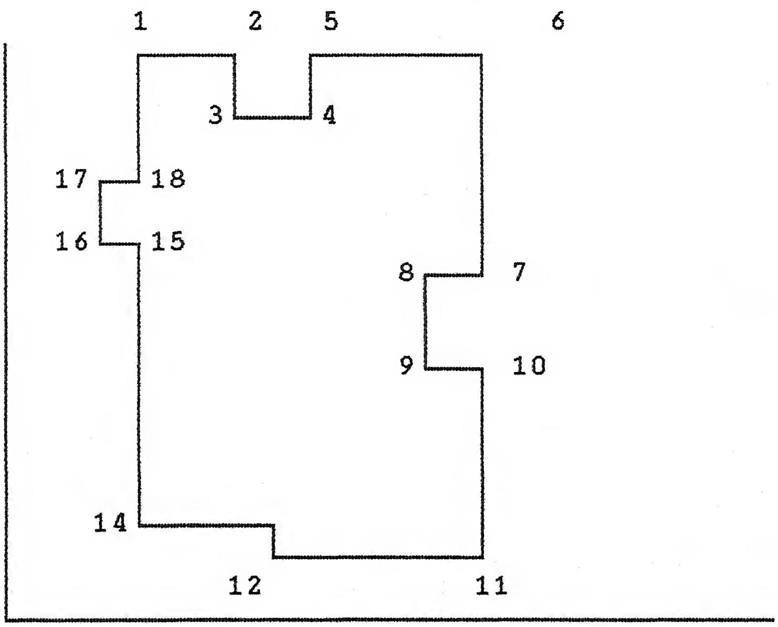


Figure 4.3 (c) : Elevation (Figure 4.3 (b) Rotated by 90 Degrees Clockwise)

Figure 4.3 : Illustrating Rotation of a Part

4.3 SEQUENCING PROCEDURE

4.3.1 IGNORING T_slots FROM A SIDE

Since the T_slots have already been defined and they will be machined last, they are eliminated from consideration of sequencing. While considering a particular side it is checked if T_slots exist on that side. If a T_slots exist and its first and last points are at the same level then all the eight points of the T_slot are ignored from that side. If they are not at the same level then a virtual point, marked as 100 in Figure 4.4, is added into that side and the remaining seven points of the T_slot are ignored from that side. With this the part drawing on the corresponding side contains only steps and slots.

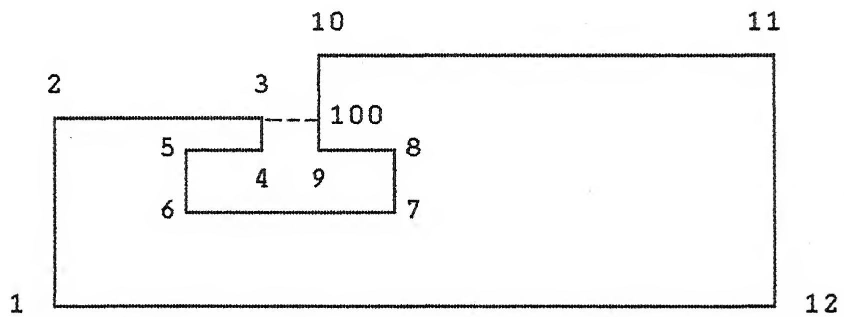


Figure 4.4 : Operation Sequencing : Consideration for T_slots

4.3.2 SEQUENCING ALGORITHM

The task now is to sequence the machining of the steps and slots on each side. Machining has to be done layer by layer from the top.

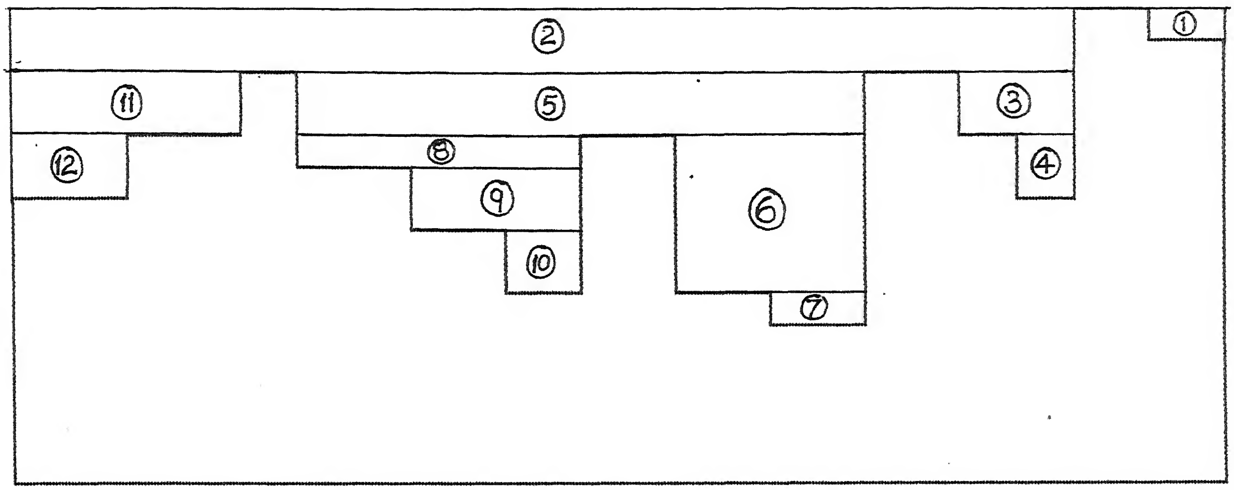


Figure 4.5 : Operation Sequencing

In sequencing the different operations, the number of tool changes is tried to be kept at a minimum. Sometimes, the steps are machined first because there is no other choice. In Figure 4.5, it can be seen that either portion 1 or portion 2 has to be machined first. Unless portion 2 is removed no slot can be machined. For machining the slots there could be many feasible ways. However, it is desirable to machine all the slots requiring the same tool at a stretch. It is assumed here that the tool starts cutting from one end (right side in Figure 4.5) and moves towards the other end(left side in Figure 4.5). For Fig. 4.5, the sequence of operations would be ① , ② , ③, ④ , ⑤ , ⑥ , ⑦ , ⑧ , ⑨ , ⑩ , ⑪ and ⑫ .

In the present work, the sequencing algorithm divides all the points of a side into a number of segments, based on their Y-coordinate values. For example, in the part drawing shown in Figure 4.6, only the points having the highest Y-coordinate are considered. They divide the whole region into a number of segments. A segment is either leftmost or rightmost or is bounded by two points at the same level. Considering any segment, material has to

be removed layer by layer from the top. Steps exist only in the leftmost or rightmost segments. In all other segments, steps combine with steps or other features giving rise to virtual slots. For example, in Figure 4.6, steps 4-5-6 and 9-10-11 combine together resulting in virtual slot 4-5-10-11. The implementation of the above algorithm is described in subsequent paragraphs.

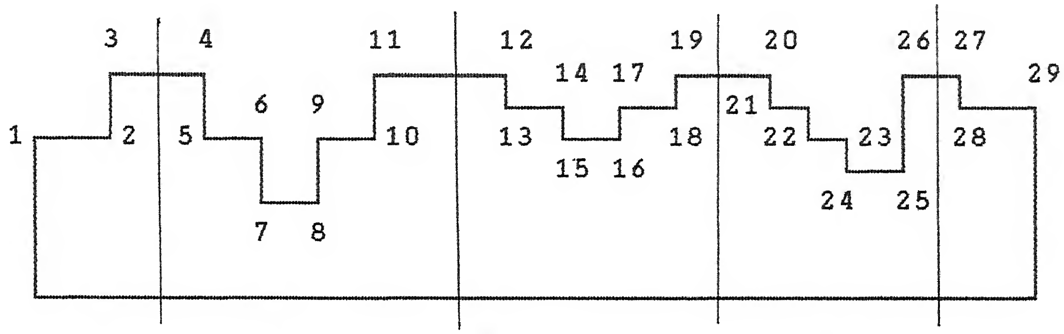


Figure 4.6 : Dividing the Side into Segments

All the points of a side are sorted on the decreasing order of their values of Y-coordinate (primary) and X-coordinate (secondary). Let the sorted array be sortpts. The following procedure finds out the steps and slots in the order they will be machined.

procedure: find_steps_and_slots(sortpts,p1,p2)

input:

sortpts, an array containing the list of points sorted on the decreasing value of their Y and X-coordinates. p1 and p2, two end points identifying a segment. All points are records of type modpt_info (defined in section 3.1).

output:

steps and slots in an array where steps and slots are arrays of records of type modpt_info

Algorithm

Step 1

Find a point (P) in between p1 and p2 such that

the Y-coordinate of point P is equal to the Y-coordinate of p1 or p2

or the Y-coordinate of point P is just smaller than the Y-coordinate of p1 or p2

and the point P is nearest to p1 or p2 whichever has a greater Y-coordinate

Step 2

If either p1 or p2 is an end point of the side then output p1, P,and p2 as step otherwise output this as slot.

Step 3

If P and p1 have different Y-coordinates

then if p1.x = (previous of point p1).x

then take p1 and P

else take previous of point p1 and P

and call steps 1 through 3 recursively

else

take a pair of points having same Y-coordinates procedure recursively with the two points as parameters to find out steps and slots in that segment

Step 4

If any more segment left

then if p2 and P have different Y-coordinates

then call the procedure recursively by taking

point P and successor of point p2

else call the procedure recursively by taking

point P and point p2

4.3.3 SEQUENCING THE DRILLING OF HOLES

In sequencing the drilling of holes, two factors are considered: (i) the number of tool changes should be minimum and (ii) the distance of tool travel should be minimum.

To keep the number of tool changes at a minimum, all holes having the same diameter first are drilled at a stretch. To keep the distance of tool travel at a minimum, the nearest neighbourhood approach is applied. This approach chooses any hole as the starting hole and at any time it chooses the nearest hole to the current one as the next hole to be drilled. This approach, of course, does not guarantee an optimum solution. However, it has been felt that on most occasions, it yields a solution which is very near to the optimum solution obtained by solving some optimization model such as travelling salesman problem.

CHAPTER V

TOOL SELECTION, NC CODE GENERATION, STATISTICS COLLECTION AND ANIMATION

This chapter describes the selection of tools for different machining operations of the part drawing, generating NC codes for them, collecting machining statistics for those operations and verifying the NC codes.

5.1 TOOL SELECTION

Selecting a tool for any machining operation depends upon some or all of the following factors:

(a) The workpiece material

(b) The machine tool

(c) The amount of material to be removed

(d) Surface finish to be achieved

(e) Tolerance to be achieved

In the present work four types of machining operations have been considered: end milling, slot cutting, T_slot cutting and drilling. A tool database containing the relevant information regarding the tools for these four operations is created. In the present implementation, the selection of tool is done on the basis of length and width of the material to be removed in case of steps and slots. For holes and T_slots the tool database contains the tool codes, based on the diameter of the hole to be drilled,and

width and depth of T_slot to be cut. The tool selection module helps to collect statistics about the tools.

5.2 NC CODES

The preparation of a set of instructions to carry out the machining of a workpiece on an NC or CNC machine is called NC part programming. The data necessary for producing a part on an NC or CNC machine are:

(a) General specifications:

Machine tool, tools, workpiece material etc.

(b) Workpiece geometry:

Dimensions (length, width, height etc.), segment shape(linear,circular or parabolic), diameter of holes to be drilled etc.

(c) Machining parameters:

cutting feed and speed (these parameters depend upon the surface quality, tolerance and type of cutting tool and workpiece material).

(d) Data determined by the part program:

Cutting direction, tool change etc.

(e) NC system characteristics:

Acceleration and deceleration intervals etc.

NC programming is of two types: (1) Manual and (2) Computer assisted.

In manual programming the data required for machining a part is written in a standard format, on the manuscript known as part program. The part program contains a list of instructions to produce the part. Each horizontal line of the program is called a

block. Manual programming is used when the part to be produced is rather simple because, it involves a lot of calculations. This method is economically justified for programming tasks like drilling, straight line cutting etc.

For complicated parts, general purpose computers are used as an aid to NC programming and several special purpose languages have been developed for part programming. The most well known and comprehensive one is APT (Automatically Programmed Tools) system, which has an English-like language input, simplifying the part programmer's job.

In the present work , the manual programming approach is adopted for NC code generation. A brief review of the manual programming is presented in the next section.

5.2.1 MANUAL PROGRAMMING

The manual part program (Koren & Ben-uri, [4]) is written on a manuscript. Each line (block) of the manuscript contains data to move the cutting tool from one point to another point, including all machining instructions that should be executed in this path. A typical full line-format is as follows:

n102,g05,x-1000,y5000,f315,s717,t65432,m03 (EB)

where the number following n is a sequential number; g - preparatory function; x and y - dimensional words; f - feed rate code; s - spindle speed code; t - tool code; m - miscellaneous function and EB is the character sign for end of block. The various functions are explained in the subsequent paragraphs.

(1) <u>Sequence</u> <u>Number</u> (n): The purpose of the sequence no. is to identify the block number of a part program. During actual machining this number guides the part programmer in comparing the actual performance with the programmed performance.

(2) <u>Preparatory Function</u> (g): The preparatory function prepares the machine to be ready to perform a specific mode of operation. For example, g90 tells the machine control unit to follow an absolute dimension mode where as g91 tells to follow the incremental dimension mode.

(3) <u>Dimension Words</u>: Dimension words are classified as follows:

(a) Distance dimension words, whose address characters (first letter of the word) are:

x,y,z for primary X,Y,Z (respectively) motion dimension

u,v,w for secondary motion dimension parallel to X,Y,Z axes respectively.

p,q,r for tertiary motion dimension parallel to X,Y,Z axes respectively.

(b) Circular dimension and thread cutting dimension words use the same address characters.

(4) <u>Feed Function</u> (f): Feed rates of linear or circular motion are independent of spindle speed and are expressed by inches/min or by mm/min rather than by inches/rev or by mm/rev as in conventional machines.

There are several methods for expressing the feed-rate number (FRN) of which one method is described below.

Inverse Time Code method- In this method the feed-rate number (FRN) code is inversely proportional to the interpolation time in

minutes. The FRN is expressed by a four digit number ranging from 0001 to 9999. This corresponds to ten minutes and 0.06 seconds respectively.

For linear motion FRN is given by

$$FRN = 10 \times \frac{\text{Velocity along path of motion (mm/min)}}{\text{length of tool path (mm)}}$$

and for circular motion it is given by

$$FRN = 10 \times \frac{\text{Velocity along arc (mm/min)}}{\text{arc radius (mm)}}$$

(5) Spindle Speed Function (s): The spindle speed function is programmed in revolution per minute and expressed by a three digit code. It is usually calculated by the magic three code method which is described in the next paragraph.

i) The speed in rpm is rounded off to two digits accuracy

ii) These two digits give the second and third digits of the code

iii) The speed is written in form of the biggest decimal fraction multiplied by power of ten

iv) The first digit of the code has a value of three greater than the power of ten.

For example, 1790 is rounded off to 1800 and written as 0.18×10^4 and the code number will be 718.

(6) Tool Function (t): The tool function can have a maximum of five digits. Each tool must have a different code number.

(7) The Miscellaneous Function (m): The miscellaneous function word consists of two digits. This function pertains to auxiliary

informations like spindle command, coolant OFF/ON, tool change, clamp, unclamp etc.

5.2.2 PRESENT IMPLEMENTATION

In the present work, NC part programming (also known as gm codes) has been generated for straight-line cuts and drilling operations. It is assumed that the tool can cut only in one direction. The cutting is done from right to left for a step in the right side, from left to right for a step in the left side and from right to left for a slot. The speed, feed and depth of cut are specified by the user. The different g and m codes that have been used and their functions are listed in the following table.

CODE	FUNCTION
g00	point-to-point positioning
g01	linear interpolation
g17	plane selection, XY horizontal
g90	absolute dimension programming
g91	incremental dimension programming
m06	tool change with automatic return traverse
m02	program end

The present system can be interfaced with the process planning system developed by Aherrao [5] which determines the machining data optimally for certain milling operations.

5.3 STATISTICS COLLECTION

Statistics collection about tools is an an important activity for tool management. Information such as how many tools

chosen for implementation and further discussion.

Animation is possible through the use of graphics facilities in the computer. A brief description of the basics of Computer Graphics (Kelly-Bootle. S., [6]) is presented in the next section.

5.4.1 BASICS OF COMPUTER GRAPHICS

Every computer, providing graphics facilities contains some display memory (or video memory) which is physically distinct from the main RAM (Random Access Memory). The monitor or the Cathode Ray Tube (CRT) display unit is fed from the display memory. The transfer from display memory to screen, in the form of a raster scan, is repeated sixty times per second giving the illusion of a steady display. The mapping of display memory to positions on the screen is of two types: Text mapping and Graphics mapping.

<u>Text Mapping</u>

In text mapping, two adjacent bytes in display memory map to a region of the screen large enough to display a single character. One byte specifies the character, and the other byte controls the character's attributes (intensity, underlining, reverse video, suppression, blinking and color).

<u>Graphics Mapping</u>

The Graphics map is fundamentally different from the text map. In the graphics mode, the screen is conceptually divided into a fine mesh of individual dots or pixels (picture elements). The dots are not, of course, Euclidean points, although it is convenient to treat them as such.

Any pixel can be suppressed or eliminated, which is the foundation for creating graphical images. The resolution of the display is expressed in terms of the number of pixels per line and the number of pixels per column. A 640 x 350 display, for example, is considered high resolution and will allow more detailed images than a 160 x 200 low resolution display.

Resolution is important for obtaining smooth curves and diagonal lines. Also, since monitor screens are wider than their height (or vice versa) and/or have pixels that are not perfect squares, there exists the problem of aspect ratio. This is noticeable while drawing exact squares and circles. If a pixel has an aspect ratio of 4:3 (4 units high and 3 units wide) then a circle would look like an ellipse, with the major axis in the Y-direction and minor axis in the X-direction. To overcome this problem, certain scaling corrections ave to be applied in one direction or the other.

The representation of a pixel requires 1 to 16 bits. The actual number of bits depends upon the range of colors available.

In the graphics mode, (X,Y) refers to a pixel rather than to a character position as in the case of the text mode.

5.4.2 PRESENT IMPLEMENTATION

The present implementation has been done on a PC with an EGA monitor. The resolution of such a monitor is 640 x 350. Following are the salient features in the present implementation:

(a) The input to the animation program is the file "gm" which contains the NC codes.

(b) The unit of distance in a monitor is pixel distance. Every

dimension of the part geometry has to be represented in multiples of pixel distance. Because of this, a drawing cannot be represented accurately on the screen. However when the actual pixel distance is rounded off then the error remains limited to half a pixel distance.

(c) First, the extra material is removed from the sides 1,2,3 & 4, in that order. Then each side, from side 1 to side 4 is oriented to face top and all features on the corresponding side are machined. After the machining on a side is over, the user is expected to change to other side in the specified order (1->2->3->4) by pressing any key.

(d) Animation proceeds in the order of the sequence of operations for a side.

(e) As mentioned in section 5.2.2, the animation of the milling is done from right to left for a step in the right side, from left to right for a step in the left side and from right to left for a slot.

(f) The tool path movement is shown in the plan view and the material removal is shown in the elevation.

(g) A person observing the animation of a part on the screen cannot distinguish between, say 1000.01mm and 1000.49mm. So the X,Y,Z coordinates of the center of the tool, just before the start and just after the end of an operation is shown on the screen. These information are also stored in a file CUT.OUT.

(h) The total cutting time required for machining a side is displayed on the screen.

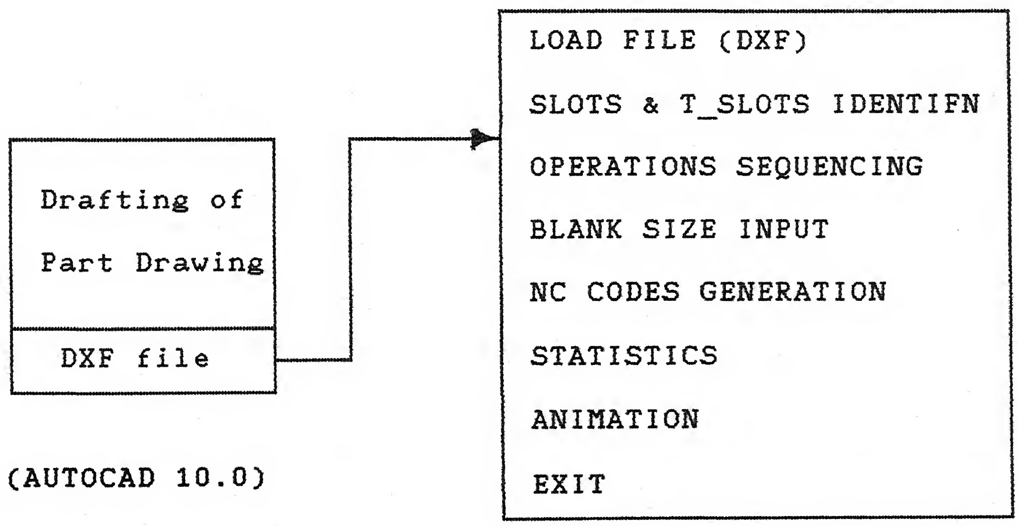
CHAPTER VI

AN ILLUSTRATION

In this chapter, various attributes of the implementation are shown through an example. Following stages from drafting of the drawing to verification of the NC codes are implemented.

1. Drafting the Part Drawing.
2. Identifying slots and T_slots.
3. Dividing the Part Drawing into four different sides.
4. Sequencing the machining operations.
5. Generating the NC codes.
6. Reporting of various machining statistics.
7. Verifying the NC codes.

The steps involved in the above stages are carried out through the menu driven system as depicted in subsequent paragraphs.



CAM

The plan view of the part drawing considered is shown in Figure 6.1

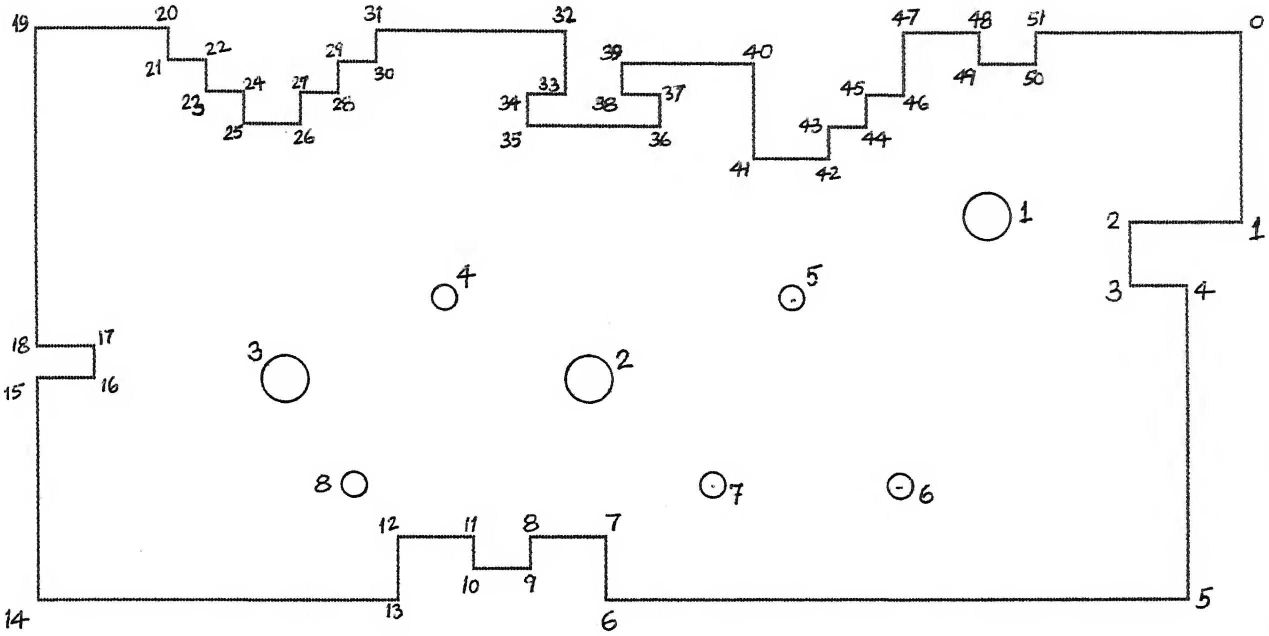


Figure 6.1 : An Illustration

The various steps are:

Step 1 : (Drafting the Part Drawing)

The part drawing is drafted in Autocad version 10.0 as two polylines, one consisting of points 14,15,16,....0 and the other consisting of points 14,13,12,...0. The coordinates of the points are given in the table overleaf.

Point No.	X Coordinate	Y Coordinate
0	919.90	657.53
1	919.90	328.76
2	830.16	328.76
3	830.16	273.97
4	875.03	273.97
5	875.03	109.58
6	583.35	109.58
7	583.35	164.38
8	538.48	164.38
9	538.48	136.98
10	493.60	136.98
11	493.60	164.38
12	448.73	164.38
13	448.73	109.58
14	224.36	109.58
15	224.36	328.76
16	269.24	328.76
17	269.24	383.56
18	224.36	383.56
19	224.36	657.53
20	314.11	657.53
21	314.11	630.13
22	336.55	630.13
23	336.55	602.73
24	358.98	602.73
25	358.98	575.34

Point No.	X Coordinate	Y Coordinate
26	403.86	575.34
27	403.86	602.73
28	426.29	602.73
29	426.29	630.13
30	448.73	630.13
31	448.73	657.53
32	516.04	657.53
33	516.04	602.73
34	493.60	602.73
35	493.60	575.34
36	560.91	575.34
37	560.91	602.73
38	538.48	602.73
39	538.48	630.13
40	628.22	630.13
41	628.22	547.94
42	673.10	547.94
43	673.10	575.34
44	695.53	575.34
45	695.53	602.73
46	717.97	602.73
47	717.97	657.53
48	807.72	657.53
49	807.72	630.13
50	852.59	630.13
51	852.59	657.53

The circles are drafted by specifying the coordinates of the center point and radius of each of them. These information are listed in the following table.

Circle No.	Center X-coordinate	Center Y-coordinate	Radius
1	807.72	465.75	54.79
2	560.91	356.16	54.79
3	336.55	356.16	54.79
4	516.04	465.75	27.39
5	650.66	465.75	27.39
6	650.66	219.17	27.39
7	516.04	219.17	27.39
8	336.55	219.17	27.39

The two polylines are exploded, giving rise to several lines and each line is given an individual identity.

Step 2 : (Identifying Slots and T_slots)

The procedure sequences_points (described in section 3.1) results in the following sequence of the points: 51,50,49,...1,0. Since this sequence is anti-clockwise, the order of the points is reversed and the resulting clockwise sequence obtained is 0,1,2, ... ,51. The slots and T_slots are identified through procedures Recognize_slot and Recognize_Tslot (described in section 3.2) and they are:

Slots : 1-2-3-4
6-7-8-9
10-11-12-13
15-16-17-18
24-25-26-27
34-35-36-37
40-41-42-43
48-49-50-51

T_slot : 32-33-34-35-36-37-38.

Since the slot 34-35-36-37 is part of a T_slot, it is deleted from the list of slots. Thus the resulting set of final slots is the following:

1-2-3-4
6-7-8-9
10-11-12-13
15-16-17-18
24-25-26-27
40-41-42-43
48-49-50-51.

Step 3 (Dividing the Part Drawing into four different sides)

The points constituting slots and T_slots are assigned side indexes based on which side they face to. For example, points 15,16,17 & 18 which form a slot facing to the left, are assigned a side index of 1. Thus after applying the procedure put_pts_into_diff_sides (described in section 3.2), the side index of the points forming slot or T_slot are as follows:

Point #	Side Index
15,16,17,18	1
24,25,26,27, 32,33,34,35, 36,37,38,39, 40,41,42,43, 48,49,50,51,	2
1,2,3,4	3
6,7,8,9, 10,11,12,13	4

To assign a side index to those points which have not been allocated a side index, the part drawing is divided into four portions. This is done by drawing a rectangle which just envelopes the part drawing and then by dividing the rectangle into four portions by its two diagonals. This is shown in figure 6.2.

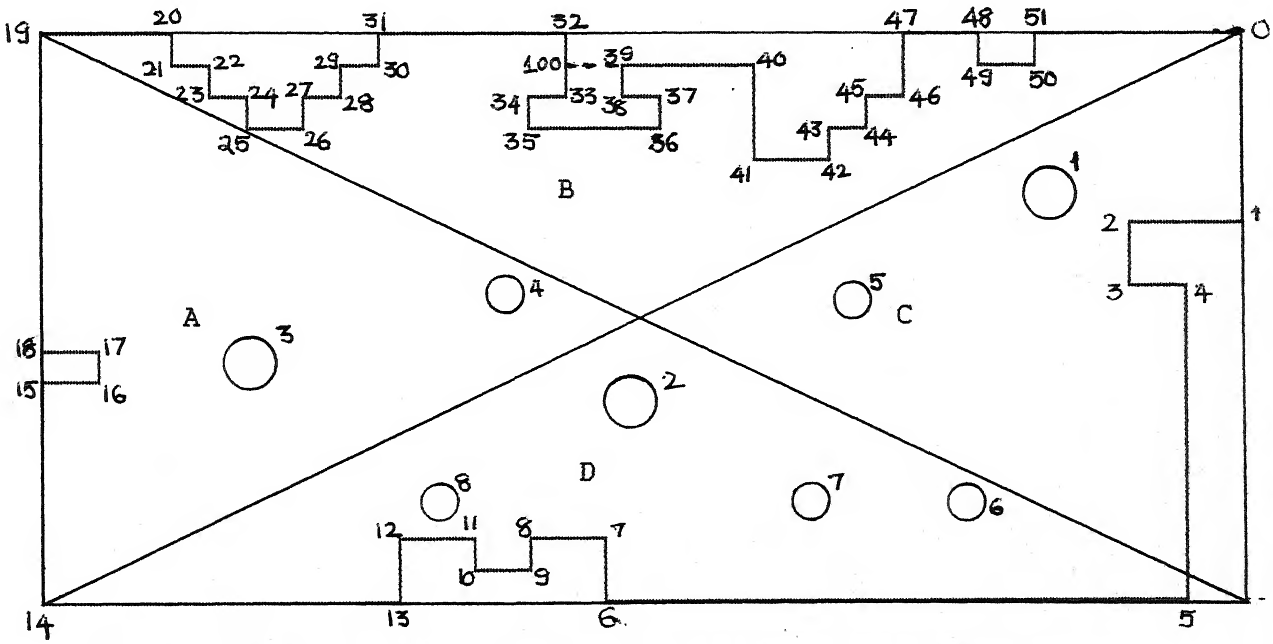


Figure 6.2 : An Illustration : Dividing into four Sides

The left-over points in portion A, B, C, D are assigned side index of 1, 2, 3 & 4 respectively. Thus the side index of the

following points is arrived at:

Point No	Side Index
19,20,21,22,23, 28,29,30,31,44,45 46,47,0	2
5,14	4

Each corner point is assigned two side indexes. For example, the corner point 14 is assigned a side index of 1 as well as 4. After assigning side index to all the points, they are segregated based on their side index and stored in four different arrays. Thus each array contains information about all the points of a particular side. The final side index of all the points is given in the following table.

Points	Side Index
14,15,16,17, 18,19	1
19,20,21,22, 23,24,25,26, 27,28,29,30, 31,32,33,34, 35,36,37,38, 39,40,41,42, 43,44,45,46, 47,48,49,50, 51,0	2
0,1,2,3,4,5	3
5,6,7,8,9,10, 11,12,13,14	4

Step 4 : (Sequencing the Machining Operations)

Applying the nearest neighbourhood approach, the sequence of the holes to be drilled is found to 8, 4, 5, 6, 7, 1, 2 and 3 (as shown in Figure 6.1)

The sequencing procedure is the same for all the four sides. Side 2 is considered here for the purpose of illustration. The T_slots are ignored from the consideration of sequencing because, they will be machined last. In this example, since the first and last points of the T_slot 32-33-34-35-36-37-38-39 have different Y-coordinate values , a virtual point, marked 100 in Figure 6.2 is stored and the points 33,34,35,36,37,38,39 are deleted from the array containing the points of side 2. The resulting points in the array are sorted on decreasing order of Y-coordinate (primary) and X-coordinate values. The list of sorted points is:

{ 0,51,48,47,32,31,20,19 },
{ 50,49,40,100,30,29,22,21 },
{ 46,45,28,27,24,23 },
{ 44,43,26,25 },
{ 42,41 }

Applying procedure Find_steps_and_slots (described in section 4.3), the following sequence of operations is obtained.

Slots :

Length From-Point	Length To-point	Depth From-Point	Depth To-point
51	48	51	50
47	32	47	100
46	40	40	46
45	40	45	44
43	40	43	42
30	20	31	30
29	22	29	28
27	24	27	26

T_slot : 32-33-34-35-36-37-38-39.

Step 5 : (Generating NC codes)

The length, width and height of the blank from which the part is to be manufactured is specified by the user as 800mm, 650mm and 100mm respectively. It is assumed that the height of the part is the same as that of the blank. The length and width of the part are 795.53mm and 547.95mm respectively. Since the length of the blank is more than the length of the part, equal amount of extra material is to be removed from both sides, right and left. Similarly, since the width of the blank is more than the width of the part, equal amount of extra material is to be removed from both top & bottom sides. NC codes are generated to remove these extra material form sides 1, 2, 3 & 4, in that order.

NC codes for all the features of the part are generated according to the plan mentioned above. The list of NC codes is is given at the end of this chapter.

Step 6 : (Reporting of Various Machining Statistics)

Before generating the NC codes for a certain operation, the tool selection is done for doing that operation. The cutting time for each operation is calculated while generating the NC codes The different tool codes, their dimensions and cutting time for each of them is listed in the file 'TOOLS.OUT'which is given at the end of this chapter.

Step 7 : (Verifying the NC codes)

The verification of the NC codes is done by animating the machining of the part on the screen of the computer. Machining in the different orientations of the part is shown in Figures 6.3 (a) through 6.3 (e). The X,Y,Z coordinates of the tool center before and after each operation are stored in file 'CUT.OUT' which is given at the end of this chapter.

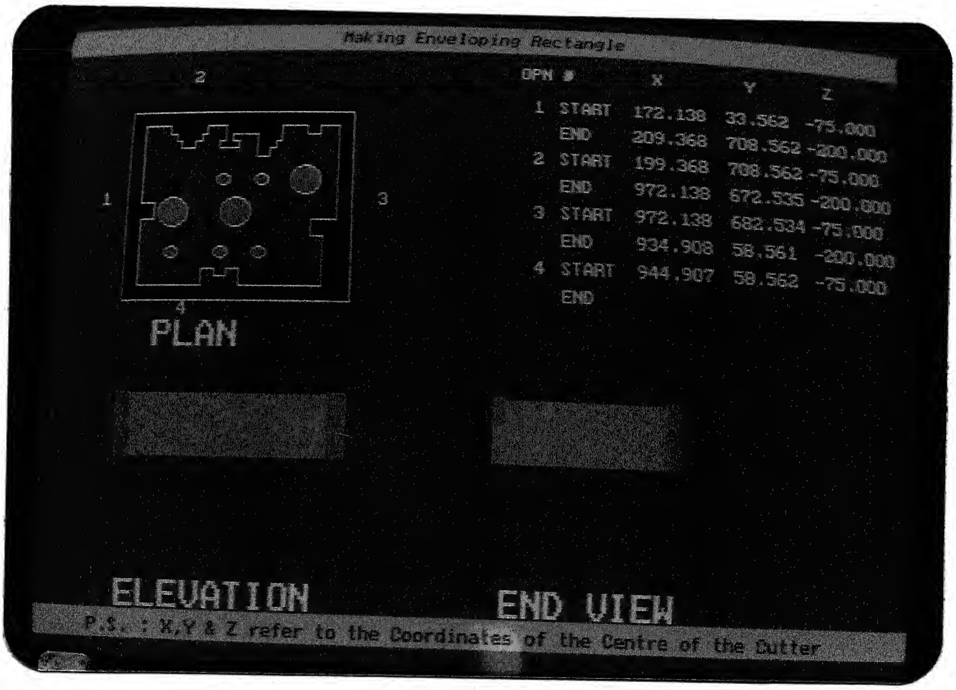


Figure 6.3 (a) : Making Eneveloping Rectangle

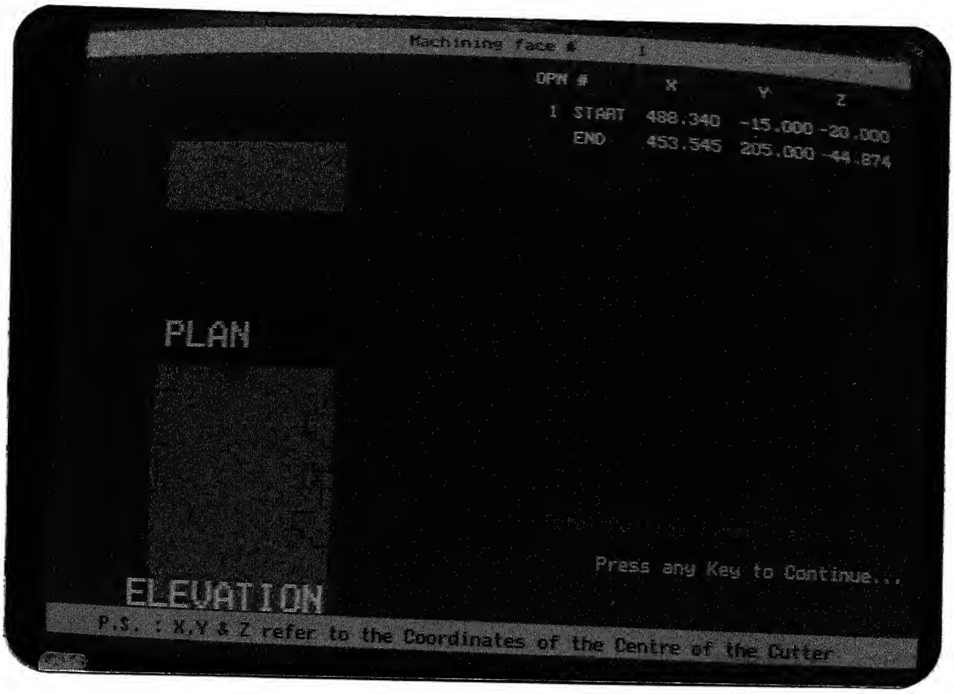


Figure 6.3 (b) : Machining Face # 1

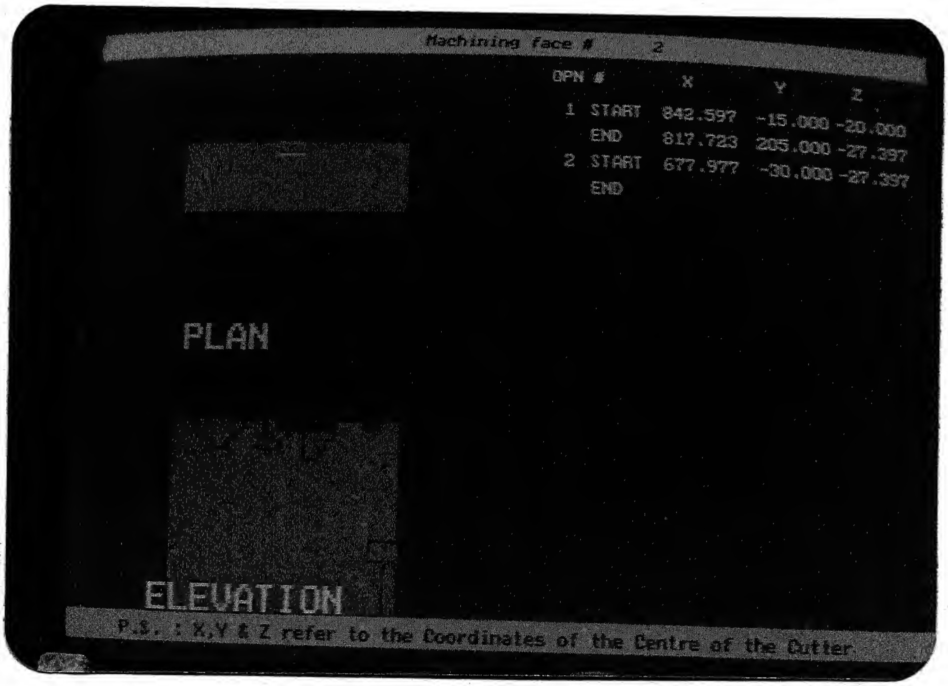


Figure 6.3 (c) : Machining Face # 2

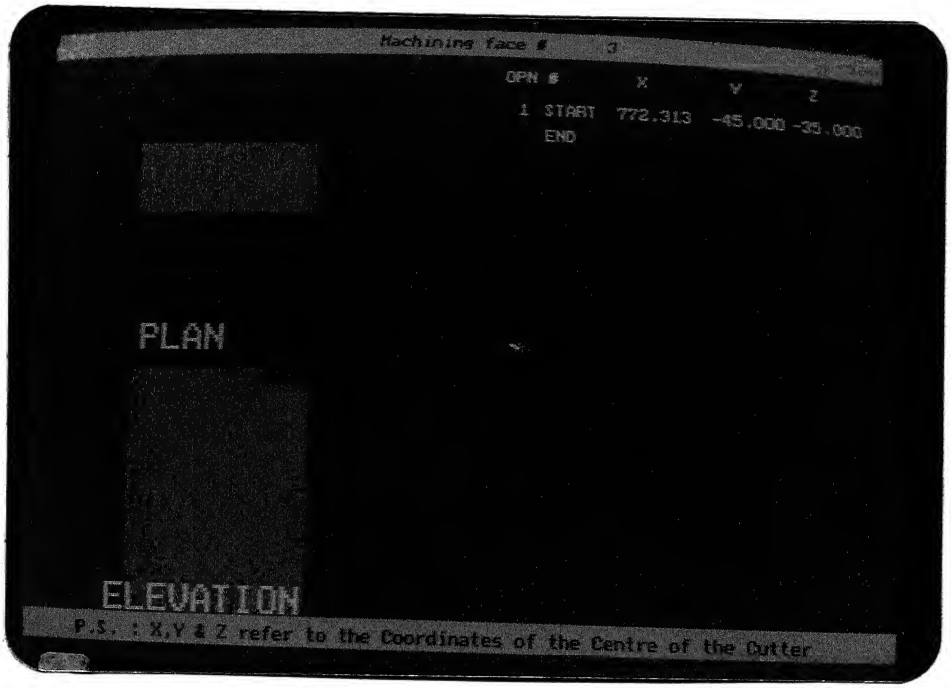


Figure 6.3 (d) : Machining Face # 3

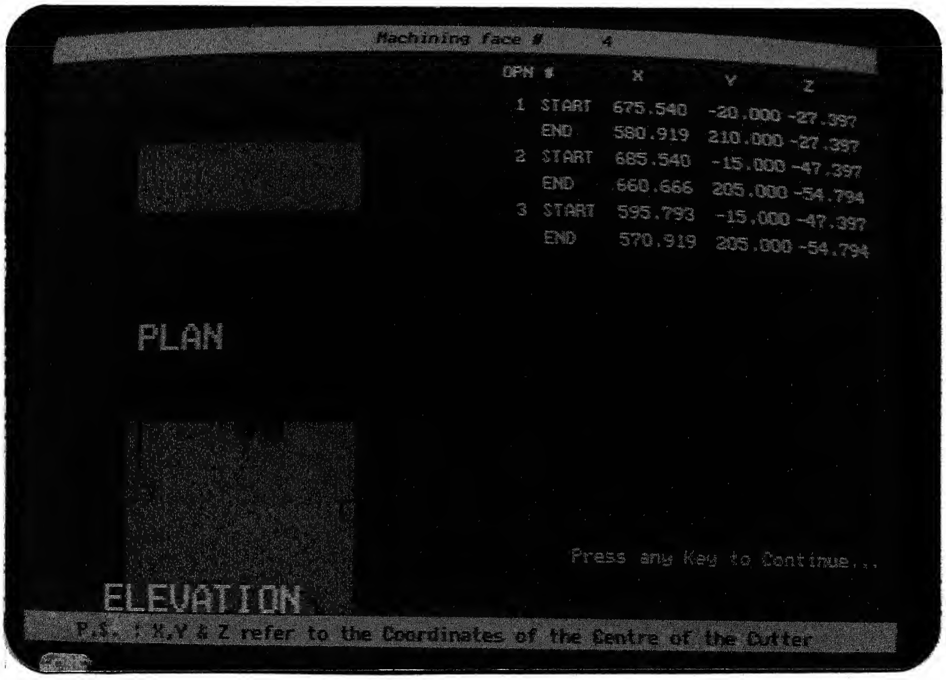


Figure 6.3 (e) : Machining Face # 4

```
(********** G & M Codes for Face 1
(**RAD   20.000
N100 G17 T2 M6
N101  f5.000
N102 G90 S500 M3
N103 G0 Z0.000
N104 G0 X142.138
N105 G0 Y28.562
N106 G91
N107 G0 Z-100.000
N108 G0 X30.000
N109 G1 Y680.000
N110 G0 X-10.000
N111 G0 Y-680.000
N112 G0 X30.000
N113 G1 Y680.000
N114 G0 X-10.000
N115 G0 Y-680.000
N116 G0 X22.230
N117 G1 Y680.000
N118 G0 X-10.000
N119 G0 Y-680.000
N120 G0 X-52.230
(********** G & M Codes for Face 2
(**RAD  20.000
N121 G17 T2 M6
N122  f5.000
N123 G90 S500 M3
N124 G0 Z0.000
N125 G0 X194.368
N126 G0 Y738.562
N127 G91
N128 G0 Z-100.000
N129 G0 Y-30.000
N130 G1 X777.770
N131 G0 Y10.000
N132 G0 X-777.770
N133 G0 Y-30.000
N134 G1 X777.770
N135 G0 Y10.000
N136 G0 X-777.770
N137 G0 Y-21.027
N138 G1 X777.770
N139 G0 Y10.000
N140 G0 X-777.770
N141 G0 Y51.027
(********** G & M Codes for Face 3
 (**RAD  20.000
N142 G17 T2 M6
N143  f5.000
N144 G90 S500 M3
N145 G0 Z0.000
N146 G0 X1002.138
```

```
N147 G0 Y687.534
N148 G91
N149 G0 Z-100.000
N150 G0 X-30.000
N151 G1 Y-628.973
N152 G0 X10.000
N153 G0 Y628.973
N154 G0 X-30.000
N155 G1 Y-628.973
N156 G0 X10.000
N157 G0 Y628.973
N158 G0 X-22.230
N159 G1 Y-628.973
N160 G0 X10.000
N161 G0 Y628.973
N162 G0 X52.230
(********** G & M Codes for Face 4
(**RAD  20.000
N163 G17 T2 M6
N164  f5.000
N165 G90 S500 M3
N166 G0 Z0.000
N167 G0 X949.907
N168 G0 Y28.562
N169 G91
N170 G0 Z-100.000
N171 G0 Y30.000
N172 G1 X-725.540
N173 G0 Y-10.000
N174 G0 X725.540
N175 G0 Y30.000
N176 G1 X-725.540
N177 G0 Y-10.000
N178 G0 X725.540
N179 G0 Y21.027
N180 G1 X-725.540
N181 G0 Y-10.000
N182 G0 X725.540
N183 G0 Y-51.027
(***********Roughcut is finished)
(**TOOLCHANGE
(**HOLES
N184 G90
(********** To machine Holes
N185 G0 Z10.000
N186 G0 X0.000
N187 G0 Y0.000
(**RAD  27.397
N188 G17 T14 M6
N189  f5.000
(*********** NC Codes for Hole  7
N190 G0 Z10.000
N191 G0 X336.552
N192 G0 Y219.178
N193 G1 Z-110.000
```

```
(*********** NC Codes for Hole  3
N194 G0 Z10.000
N195 G0 X516.046
N196 G0 Y465.753
N197 G1 Z-110.000
(*********** NC Codes for Hole  4
N198 G0 Z10.000
N199 G0 X650.666
N200 G0 Y465.753
N201 G1 Z-110.000
(*********** NC Codes for Hole  5
N202 G0 Z10.000
N203 G0 X650.666
N204 G0 Y219.178
N205 G1 Z-110.000
(*********** NC Codes for Hole  6
N206 G0 Z10.000
N207 G0 X516.046
N208 G0 Y219.178
N209 G1 Z-110.000
(********** To machine Holes
N210 G0 Z10.000
N211 G0 X0.000
N212 G0 Y0.000
(**RAD  54.794
N213 G17 T15 M6
N214  f5.000
(*********** NC Codes for Hole  0
N215 G0 Z10.000
N216 G0 X807.724
N217 G0 Y465.753
N218 G1 Z-110.000
(*********** NC Codes for Hole  1
N219 G0 Z10.000
N220 G0 X560.919
N221 G0 Y356.164
N222 G1 Z-110.000
(*********** NC Codes for Hole  2
N223 G0 Z10.000
N224 G0 X336.552
N225 G0 Y356.164
N226 G1 Z-110.000
N227 G0 Z10.000
(********* Steps,Slots & Tslots for Face no ) 1
(**FACECHANGE
(*********** NC Codes for slot )
(**WIDTH   20.000
N228 G90
N229 G0 Z10.000
N230 G0 X0.000
N231 G17 T5 M6
N232  f5.000
N233 G0 Y-15.000
N234 G0 X488.340
N231 G91
```

```
N235 G0 Z-0.000
N236 G0 Z-20.000
N237 G0 Z-10.000
N238 G1 Y120.000
N239 G0 Z10.000
N240 G0 Y-120.000
N241 G0 X-20.000
N242 G0 Z-10.000
N243 G1 Y120.000
N244 G0 Z10.000
N245 G0 Y-120.000
N246 G0 X-14.795
N247 G0 Z-10.000
N248 G1 Y120.000
N249 G0 Z10.000
N250 G0 Y-120.000
N251 G0 X-0.000
N252 G0 X34.795
N253 G0 Z-20.000
N254 G0 Z-10.000
N255 G1 Y120.000
N256 G0 Z10.000
N257 G0 Y-120.000
N258 G0 X-20.000
N259 G0 Z-10.000
N260 G1 Y120.000
N261 G0 Z10.000
N262 G0 Y-120.000
N263 G0 X-14.795
N264 G0 Z-10.000
N265 G1 Y120.000
N266 G0 Z10.000
N267 G0 Y-120.000
N268 G0 X-0.000
N269 G0 X34.795
N270 G0 Z-4.874
N271 G0 Z-10.000
N272 G1 Y120.000
N273 G0 Z10.000
N274 G0 Y-120.000
N275 G0 X-20.000
N276 G0 Z-10.000
N277 G1 Y120.000
N278 G0 Z10.000
N279 G0 Y-120.000
N280 G0 X-14.795
N281 G0 Z-10.000
N282 G1 Y120.000
N283 G0 Z10.000
N284 G0 Y-120.000
N285 G0 X-0.000
N286 G0 X34.795
(********* Steps,Slots & Tslots for Face no ) 2
(**FACECHANGE
(*********** NC Codes for slot )
```

```
(**WIDTH   20.000
N287 G90
N288 G0 Z10.000
N289 G0 X0.000
N290 G17 T5 M6
N291  f5.000
N292 G0 Y-15.000
N293 G0 X842.597
N290 G91
N294 G0 Z-0.000
N295 G0 Z-20.000
N296 G0 Z-10.000
N297 G1 Y120.000
N298 G0 Z10.000
N299 G0 Y-120.000
N300 G0 X-20.000
N301 G0 Z-10.000
N302 G1 Y120.000
N303 G0 Z10.000
N304 G0 Y-120.000
N305 G0 X-4.874
N306 G0 Z-10.000
N307 G1 Y120.000
N308 G0 Z10.000
N309 G0 Y-120.000
N310 G0 X-0.000
N311 G0 X24.874
N312 G0 Z-7.397
N313 G0 Z-10.000
N314 G1 Y120.000
N315 G0 Z10.000
N316 G0 Y-120.000
N317 G0 X-20.000
N318 G0 Z-10.000
N319 G1 Y120.000
N320 G0 Z10.000
N321 G0 Y-120.000
N322 G0 X-4.874
N323 G0 Z-10.000
N324 G1 Y120.000
N325 G0 Z10.000
N326 G0 Y-120.000
N327 G0 X-0.000
N328 G0 X24.874
(*********** NC Codes for slot )
(**WIDTH   150.000
N329 G90
N330 G0 Z10.000
N331 G0 X0.000
N332 G17 T8 M6
N333  f0.667
N334 G0 Y-47.500
N335 G0 X642.977
N332 G91
N336 G0 Z-0.000
```

```
N337 G0 Z-27.397
N338 G0 Z-10.000
N339 G1 Y185.000
N340 G0 Z10.000
N341 G0 Y-185.000
N342 G0 X-51.931
N343 G0 Z-10.000
N344 G1 Y185.000
N345 G0 Z10.000
N346 G0 Y-185.000
N347 G0 X-0.000
N348 G0 X51.931
(*********** NC Codes for slot )
(**WIDTH   50.000
N349 G90
N350 G0 Z10.000
N351 G0 X0.000
N352 G17 T6 M6
N353  f2.000
N354 G0 Y-22.500
N355 G0 X692.977
N352 G91
N356 G0 Z-27.397
N357 G0 Z-27.397
N358 G0 Z-10.000
N359 G1 Y135.000
N360 G0 Z10.000
N361 G0 Y-135.000
N362 G0 X-39.747
N363 G0 Z-10.000
N364 G1 Y135.000
N365 G0 Z10.000
N366 G0 Y-135.000
N367 G0 X-0.000
N368 G0 X39.747
(*********** NC Codes for slot )
(**WIDTH   50.000
N369 G90
N370 G0 Z10.000
N371 G0 X0.000
N372 G17 T6 M6
N373  f2.000
N374 G0 Y-22.500
N375 G0 X670.540
N372 G91
N376 G0 Z-54.794
N377 G0 Z-27.397
N378 G0 Z-10.000
N379 G1 Y135.000
N380 G0 Z10.000
N381 G0 Y-135.000
N382 G0 X-17.310
N383 G0 Z-10.000
N384 G1 Y135.000
N385 G0 Z10.000
```

```
N386 G0 Y-135.000
N387 G0 X-0.000
N388 G0 X17.310
(*********** NC Codes for slot )
(**WIDTH   20.000
N389 G90
N390 G0 Z10.000
N391 G0 X0.000
N392 G17 T5 M6
N393  f5.000
N394 G0 Y-15.000
N395 G0 X663.103
N392 G91
N396 G0 Z-82.192
N397 G0 Z-20.000
N398 G0 Z-10.000
N399 G1 Y120.000
N400 G0 Z10.000
N401 G0 Y-120.000
N402 G0 X-20.000
N403 G0 Z-10.000
N404 G1 Y120.000
N405 G0 Z10.000
N406 G0 Y-120.000
N407 G0 X-4.874
N408 G0 Z-10.000
N409 G1 Y120.000
N410 G0 Z10.000
N411 G0 Y-120.000
N412 G0 X-0.000
N413 G0 X24.874
N414 G0 Z-7.397
N415 G0 Z-10.000
N416 G1 Y120.000
N417 G0 Z10.000
N418 G0 Y-120.000
N419 G0 X-20.000
N420 G0 Z-10.000
N421 G1 Y120.000
N422 G0 Z10.000
N423 G0 Y-120.000
N424 G0 X-4.874
N425 G0 Z-10.000
N426 G1 Y120.000
N427 G0 Z10.000
N428 G0 Y-120.000
N429 G0 X-0.000
N430 G0 X24.874
(*********** NC Codes for slot )
(**WIDTH   100.000
N431 G90
N432 G0 Z10.000
N433 G0 X0.000
N434 G17 T7 M6
N435  f1.000
```

```
N436 G0 Y-35.000
N437 G0 X398.735
N434 G91
N438 G0 Z-0.000
N439 G0 Z-27.397
N440 G0 Z-10.000
N441 G1 Y160.000
N442 G0 Z10.000
N443 G0 Y-160.000
N444 G0 X-34.621
N445 G0 Z-10.000
N446 G1 Y160.000
N447 G0 Z10.000
N448 G0 Y-160.000
N449 G0 X-0.000
N450 G0 X34.621
(*********** NC Codes for slot )
(**WIDTH   50.000
N451 G90
N452 G0 Z10.000
N453 G0 X0.000
N454 G17 T6 M6
N455  f2.000
N456 G0 Y-22.500
N457 G0 X401.299
N454 G91
N458 G0 Z-27.397
N459 G0 Z-27.397
N460 G0 Z-10.000
N461 G1 Y135.000
N462 G0 Z10.000
N463 G0 Y-135.000
N464 G0 X-39.747
N465 G0 Z-10.000
N466 G1 Y135.000
N467 G0 Z10.000
N468 G0 Y-135.000
N469 G0 X-0.000
N470 G0 X39.747
(*********** NC Codes for slot )
(**WIDTH   20.000
N471 G90
N472 G0 Z10.000
N473 G0 X0.000
N474 G17 T5 M6
N475  f5.000
N476 G0 Y-15.000
N477 G0 X393.862
N474 G91
N478 G0 Z-54.794
N479 G0 Z-20.000
N480 G0 Z-10.000
N481 G1 Y120.000
N482 G0 Z10.000
N483 G0 Y-120.000
```

```
N484 G0 X-20.000
N485 G0 Z-10.000
N486 G1 Y120.000
N487 G0 Z10.000
N488 G0 Y-120.000
N489 G0 X-4.873
N490 G0 Z-10.000
N491 G1 Y120.000
N492 G0 Z10.000
N493 G0 Y-120.000
N494 G0 X-0.000
N495 G0 X24.873
N496 G0 Z-7.397
N497 G0 Z-10.000
N498 G1 Y120.000
N499 G0 Z10.000
N500 G0 Y-120.000
N501 G0 X-20.000
N502 G0 Z-10.000
N503 G1 Y120.000
N504 G0 Z10.000
N505 G0 Y-120.000
N506 G0 X-4.873
N507 G0 Z-10.000
N508 G1 Y120.000
N509 G0 Z10.000
N510 G0 Y-120.000
N511 G0 X-0.000
N512 G0 X24.873
(*********** NC Codes for Tslot
(**WIDTH   20.000
N513 G90
N514 G0 Z10.000
N515 G0 X0.000
N516 G17 T5 M6
N517  f5.000
N518 G0 Y-15.000
N519 G0 X528.482
N516 G91
N520 G0 Z-27.397
N521 G0 Z-20.000
N522 G0 Z-10.000
N523 G1 Y120.000
N524 G0 Z10.000
N525 G0 Y-120.000
N526 G0 X-2.437
N527 G0 Z-10.000
N528 G1 Y120.000
N529 G0 Z10.000
N530 G0 Y-120.000
N531 G0 X-0.000
N532 G0 X2.437
N533 G0 Z-20.000
N534 G0 Z-10.000
N535 G1 Y120.000
```

```
N536 G0 Z10.000
N537 G0 Y-120.000
N538 G0 X-2.437
N539 G0 Z-10.000
N540 G1 Y120.000
N541 G0 Z10.000
N542 G0 Y-120.000
N543 G0 X-0.000
N544 G0 X2.437
N545 G0 Z-14.794
N546 G0 Z-10.000
N547 G1 Y120.000
N548 G0 Z10.000
N549 G0 Y-120.000
N550 G0 X-2.437
N551 G0 Z-10.000
N552 G1 Y120.000
N553 G0 Z10.000
N554 G0 Y-120.000
N555 G0 X-0.000
N556 G0 X2.437
(**WIDTH   67.310
(**THICKNESS  27.397
N557 G90
N557 G17 T9 M6
N558  f1.486
N559 G0 Y-23.699
N560 G0 X527.264
N561 G0 Z-68.493
N562  f50.000
N563 G1 Y113.699
N564 G0 Z78.493
(********* Steps,Slots & Tslots for Face no ) 3
(**FACECHANGE
(*********** NC Codes for Step  5
(**RAD   50.000
N565 G90
N566 G0 Z10.000
N567 G0 X0.000
N568 G17 T4 M6
N569  f2.000
N570 G0 Y-60.000
N571 G0 X832.313
N568 G91
N572 G0 Z-0.000
N573 G0 Z-44.874
N574 G0 X-60.000
N575 G0 Z-10.000
N576 G1 Y160.000
N577 G0 X10.000
N578 G0 Z10.000
N579 G0 Y-160.000
N580 G0 X-60.000
N581 G0 Z-10.000
N582 G1 Y160.000
```

```
N583 G0 X10.000
N584 G0 Z10.000
N585 G0 Y-160.000
N586 G0 X-60.000
N587 G0 Z-10.000
N588 G1 Y160.000
N589 G0 X10.000
N590 G0 Z10.000
N591 G0 Y-160.000
N592 G0 X-60.000
N593 G0 Z-10.000
N594 G1 Y160.000
N595 G0 X10.000
N596 G0 Z10.000
N597 G0 Y-160.000
N598 G0 X-29.178
N599 G0 Z-10.000
N600 G1 Y160.000
N601 G0 X10.000
N602 G0 Z10.000
N603 G0 Y-160.000
N604 G0 X219.178
(*********** NC Codes for slot )
(**WIDTH   20.000
N605 G90
N606 G0 Z10.000
N607 G0 X0.000
N608 G17 T5 M6
N609  f5.000
N610 G0 Y-15.000
N611 G0 X597.929
N608 G91
N612 G0 Z-44.874
N613 G0 Z-20.000
N614 G0 Z-10.000
N615 G1 Y120.000
N616 G0 Z10.000
N617 G0 Y-120.000
N618 G0 X-20.000
N619 G0 Z-10.000
N620 G1 Y120.000
N621 G0 Z10.000
N622 G0 Y-120.000
N623 G0 X-14.795
N624 G0 Z-10.000
N625 G1 Y120.000
N626 G0 Z10.000
N627 G0 Y-120.000
N628 G0 X-0.000
N629 G0 X34.795
N630 G0 Z-20.000
N631 G0 Z-10.000
N632 G1 Y120.000
N633 G0 Z10.000
N634 G0 Y-120.000
```

```
N635 G0 X-20.000
N636 G0 Z-10.000
N637 G1 Y120.000
N638 G0 Z10.000
N639 G0 Y-120.000
N640 G0 X-14.795
N641 G0 Z-10.000
N642 G1 Y120.000
N643 G0 Z10.000
N644 G0 Y-120.000
N645 G0 X-0.000
N646 G0 X34.795
N647 G0 Z-4.873
N648 G0 Z-10.000
N649 G1 Y120.000
N650 G0 Z10.000
N651 G0 Y-120.000
N652 G0 X-20.000
N653 G0 Z-10.000
N654 G1 Y120.000
N655 G0 Z10.000
N656 G0 Y-120.000
N657 G0 X-14.795
N658 G0 Z-10.000
N659 G1 Y120.000
N660 G0 Z10.000
N661 G0 Y-120.000
N662 G0 X-0.000
N663 G0 X34.795
(********* Steps,Slots & Tslots for Face no ) 4
(**FACECHANGE
(*********** NC Codes for slot )
(**WIDTH   100.000
N664 G90
N665 G0 Z10.000
N666 G0 X0.000
N667 G17 T7 M6
N668  f1.000
N669 G0 Y-35.000
N670 G0 X645.540
N667 G91
N671 G0 Z-0.000
N672 G0 Z-27.397
N673 G0 Z-10.000
N674 G1 Y160.000
N675 G0 Z10.000
N676 G0 Y-160.000
N677 G0 X-34.621
N678 G0 Z-10.000
N679 G1 Y160.000
N680 G0 Z10.000
N681 G0 Y-160.000
N682 G0 X-0.000
N683 G0 X34.621
(*********** NC Codes for slot )
```

```
(**WIDTH   20.000
N684 G90
N685 G0 Z10.000
N686 G0 X0.000
N687 G17 T5 M6
N688  f5.000
N689 G0 Y-15.000
N690 G0 X685.540
N687 G91
N691 G0 Z-27.397
N692 G0 Z-20.000
N693 G0 Z-10.000
N694 G1 Y120.000
N695 G0 Z10.000
N696 G0 Y-120.000
N697 G0 X-20.000
N698 G0 Z-10.000
N699 G1 Y120.000
N700 G0 Z10.000
N701 G0 Y-120.000
N702 G0 X-4.874
N703 G0 Z-10.000
N704 G1 Y120.000
N705 G0 Z10.000
N706 G0 Y-120.000
N707 G0 X-0.000
N708 G0 X24.874
N709 G0 Z-7.397
N710 G0 Z-10.000
N711 G1 Y120.000
N712 G0 Z10.000
N713 G0 Y-120.000
N714 G0 X-20.000
N715 G0 Z-10.000
N716 G1 Y120.000
N717 G0 Z10.000
N718 G0 Y-120.000
N719 G0 X-4.874
N720 G0 Z-10.000
N721 G1 Y120.000
N722 G0 Z10.000
N723 G0 Y-120.000
N724 G0 X-0.000
N725 G0 X24.874
(*********** NC Codes for slot )
(**WIDTH   20.000
N726 G90
N727 G0 Z10.000
N728 G0 X0.000
N729 G17 T5 M6
N730  f5.000
N731 G0 Y-15.000
N732 G0 X595.793
N729 G91
N733 G0 Z-27.397
```

```
N734 G0 Z-20.000
N735 G0 Z-10.000
N736 G1 Y120.000
N737 G0 Z10.000
N738 G0 Y-120.000
N739 G0 X-20.000
N740 G0 Z-10.000
N741 G1 Y120.000
N742 G0 Z10.000
N743 G0 Y-120.000
N744 G0 X-4.874
N745 G0 Z-10.000
N746 G1 Y120.000
N747 G0 Z10.000
N748 G0 Y-120.000
N749 G0 X-0.000
N750 G0 X24.874
N751 G0 Z-7.397
N752 G0 Z-10.000
N753 G1 Y120.000
N754 G0 Z10.000
N755 G0 Y-120.000
N756 G0 X-20.000
N757 G0 Z-10.000
N758 G1 Y120.000
N759 G0 Z10.000
N760 G0 Y-120.000
N761 G0 X-4.874
N762 G0 Z-10.000
N763 G1 Y120.000
N764 G0 Z10.000
N765 G0 Y-120.000
N766 G0 X-0.000
N767 G0 X24.874
EOF
```

TOOLS.OUT

SL. #	CODE	TYPE	NO.OF SETTINGS	DIMENSION (mm)	CUTTING TIME (Sec)
1	T2	END_CUTTER	4	40.000	101
2	T4	END_CUTTER	1	100.000	24
3	T5	SLOT_CUTTER	8	20.000	78
4	T6	SLOT_CUTTER	3	50.000	24
5	T7	SLOT_CUTTER	2	100.000	38
6	T8	SLOT_CUTTER	1	150.000	33
7	T9	TSLOT_CUTTER	1	67.310	5
8	T19	DRILL	1	54.795	9
9	T20	DRILL	1	109.589	11

P.S. DIMENSION Refers to DIAMETER for END_CUTTER & WIDTH for SLOT & TSLOT CUTTER

CUT.OUT

OPERATION #		X	Y	Z
1	START	172.138	28.562	-100.000
	END	204.368	708.562	-100.000
2	START	194.368	708.562	-100.000
	END	972.138	677.535	-100.000
3	START	972.138	687.534	-100.000
	END	939.908	58.561	-100.000
4	START	949.907	58.562	-100.000
	END	224.367	89.589	-100.000
1	START	488.340	-15.000	-20.000
	END	453.545	105.000	-44.874
1	START	842.597	-15.000	-20.000
	END	817.723	105.000	-27.397
2	START	642.977	-47.500	-27.397
	END	591.046	137.500	-27.397
3	START	692.977	-22.500	-54.794
	END	653.230	112.500	-54.794
4	START	670.540	-22.500	-82.191
	END	653.230	112.500	-82.191
5	START	663.103	-15.000	-102.192
	END	638.229	105.000	-109.589
6	START	398.735	-35.000	-27.397
	END	364.114	125.000	-27.397
7	START	401.299	-22.500	-54.794
	END	361.552	112.500	-54.794
8	START	393.862	-15.000	-74.794
	END	368.989	105.000	-82.191
9	START	528.482	-15.000	-47.397
	END	526.045	105.000	-82.191
10	START	527.264	-23.699	-68.493
	END	527.264	113.699	-68.493
1	START	772.313	-60.000	-44.874
	END	603.135	100.000	-44.874
2	START	597.929	-15.000	-64.874
	END	563.134	105.000	-89.747
1	START	645.540	-35.000	-27.397
	END	610.919	125.000	-27.397
2	START	685.540	-15.000	-47.397
	END	660.666	105.000	-54.794
3	START	595.793	-15.000	-47.397
	END	570.919	105.000	-54.794

CHAPTER VII

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

In the present system, some aspects of CAD/CAM integration i.e. identifying features (slots and T_slots), sequencing the operations and generating NC codes for those operations have been analyzed and implemented for prismatic parts with linear sweep having through holes in it.

Identifying slots and T_slots through comparison of X and Y-coordinates for a prismatic part having surfaces parallel to X and Y-axes seem to be an efficient approach. The approach can also be applied to parts having inclined surfaces. However, it may not be an efficient method to identify features having curved surfaces.

For the ease of implementation, the recognition of steps is done through a sequencing algorithm. Steps facing each other merge together giving rise to virtual slots. This merging of steps is also done through the sequencing algorithm. The algorithm is suitable when the material is to be removed layer by layer from the top. It is felt that the algorithm may have limitations for parts having curved surfaces. The nearest neighbourhood approach used for the sequencing of the drilling of holes is suitable when the number of holes is large.

Manual programming approach is used for NC codes generation incorporating g and m codes. This approach, indeed, requires the specifications of the machine and tools during the process of NC

code generation. The system, however, can accommodate a modular part programming approach comprising modules such as part geometry, machine specifications and motion commands using any of the widely available special purpose languages e.g. APT. However, these codes would require a suitable post processing software.

Validating the NC codes by animating the machining of the part on the computer monitor suffers from the problem of representing dimensions accurately on the monitor due to limited resolution. This problem is overcome by showing the X, Y and Z coordinates of the tool center immediately before and after each operation.

SUGGESTION FOR FUTURE WORK

(a) The present work is a modest attempt towards CAD/CAM integration. Further attempts can be made to enrich the concepts developed.

(b) Prismatic parts having orthogonal surfaces have been considered in the present work. Attempts can be made to extract features and generate NC codes for parts having inclined and circular surfaces.

The process planning aspect i.e. selection of material, machine, tool and determination of the machining parameters can be incorporated so as to achieve a fully integrated CAD/CAM system.

CHAPTER VIII

USERS' MANUAL

This chapter describes the step by step details of right from drafting a drawing to verifying the NC codes generated by the program.

Step 1 : Drafting a drawing

Step 1.1

Make sure that AUTOCAD version 10.0 is properly installed in a PC XT/AT/386 machine. The hardware lock should have been connected suitably to the COM1 port of the machine. If the present directory is other than the one having the files of AUTOCAD then, the path should be set to the directory containing the AUTOCAD files and type the following command from the DOS prompt.

> acad <CR>

The main menu of AUTOCAD is displayed.

Main Menu

0. Exit to AUTOCAD
1. Begin a NEW drawing
2. Edit an EXISTING drawing
3. Plot a drawing
4. Printer Plot a drawing
5. Configure AUTOCAD
6. File Utilities
7. Compile shape/font description file
8. Convert old drawing file

Enter selection:

Step 1.2

To create a new drawing, the prompt sequence will look like this:

Enter selection: 1 <CR>

Enter NAME of drawing: FILENAME <CR> (type the file name)

The AUTOCAD drawing editor appears in the screen.

Step 1.3

The default screen size is 12mm x 9mm. If the drawing is bigger than this, then the screen size is changed by the following prompt sequence:

command: limits <CR>

lower left corner<0,0>: <CR>

upper right corner<12.0,9.0>: X,Y <CR> (type the values of X,Y)

command: zoom <CR>

All/Center/Dynamic/Extents/Left/Previous/Window/<Scale(X)>: a <CR>

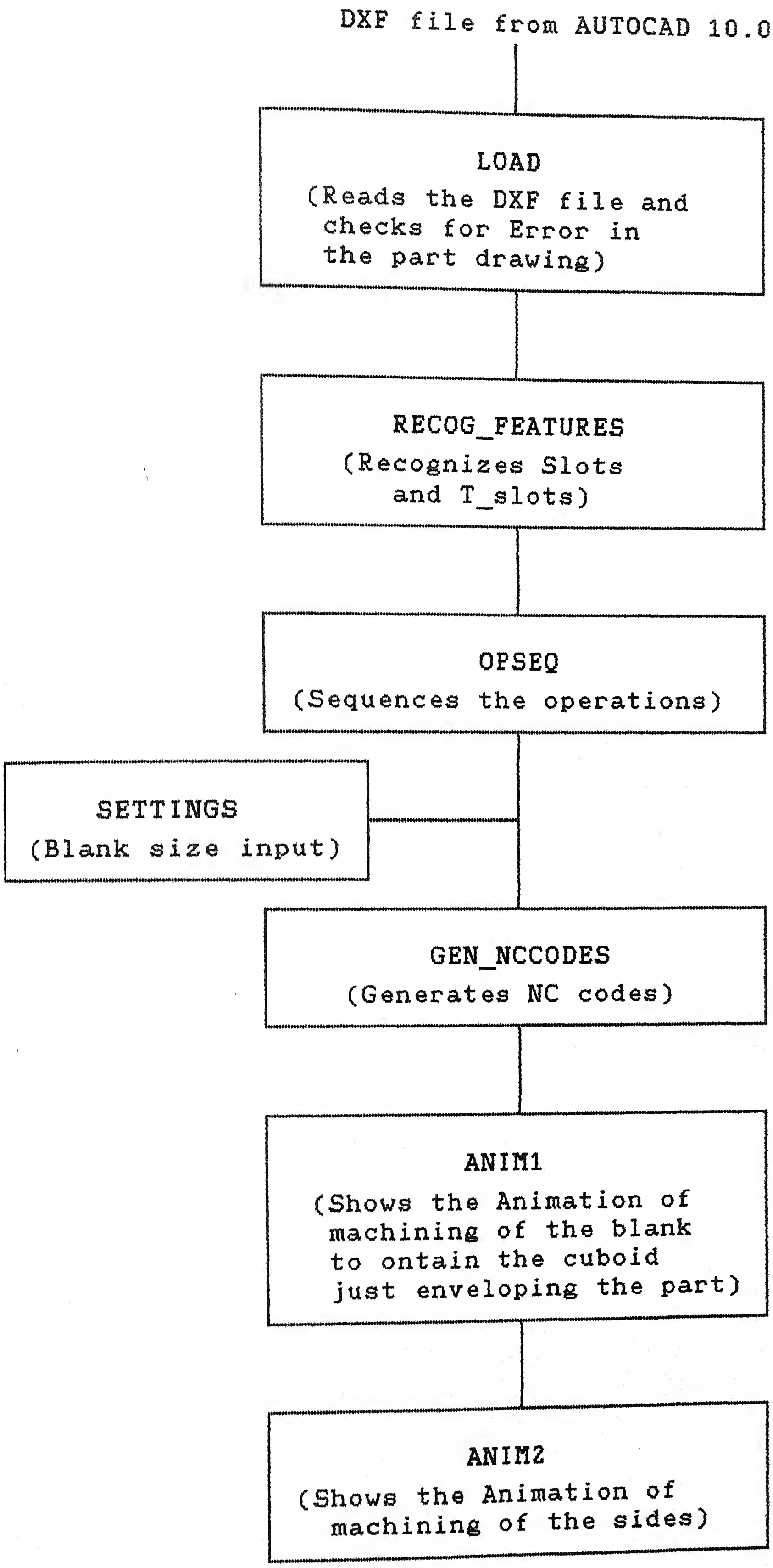
The screen size is changed from 12 mm x 9 mm to Xmm x Ymm.

Step 1.4

The drawing can be drafted by either command line or command polyline. In both the commands points can be specified either by moving the cursor (by pressing arrow keys) or by specifying the X, Y coordinates.

Step 1.5

After the drawing is drafted, it is saved by the following command:



THE SYSTEM FLOW CHART

command: save <CR>

save in file<FILENAME>: <CR>

Step 1.6

A file "carc.lsp" should exist in the current directory. This file is to be loaded by the following command:

command: (load "carc.lsp") <CR>

After a few seconds C:CAM is displayed in the lower left corner of the screen. Type

command: cam <CR>

After the command prompt appears in the screen, the original drawing is displayed.

Step 1.7

To create the DXF file of the drafted drawing, the prompt sequence is:

command: dxfout <CR>

Enter decimal places of accuracy(0 to 16)/Entities/Binary <6>:

x <CR>

(x can take any integer value between 0 to 16)

Step 1.8

The drafting is complete and DXF file is created. To quit from AUTOCAD, the prompt sequence is:

command: quit <CR>

Really want to discard all changes to drawing? y <CR>

The main menu is displayed. Type

command: 0 <CR>

It takes back to DOS environment.

Step 2 : Getting Started

The file **cam.exe** and **egavga.bgi** should exist in the directory from which the program is to be executed. The following command is to be typed from the DOS prompt.

>cam <CR>

The opening message

CAD / CAM

INTEGRATION

will be displayed. Then, pressing any key will display the main menu as shown below:

LOAD FILE (DXF)
SLOTS & T_SLOTS IDENTIFN
OPERATIONS SEQUENCING
BLANK SIZE INPUT
NC CODES GENERATION
STATISTICS
ANIMATION
EXIT

The cursor will be over "LOAD FILE (DXF)".

Step 3 : Loading a File and Viewing the drawing

Step 3.1

If the carriage return is pressed when the cursor is over "LOAD FILE (DXF)", the following message comes along with the main menu:

DXF FILE:

It is expecting the name of the DXF file. If the name of the file is test.dxf then just "test" should be typed. DXF is taken as the default extension. Carriage return should be pressed after typing the filename. On successful opening of the file, the following messages appear on the screen:

(1) A rectangle with "PLAN" written below it.

(2) "F2-Complete Drawing F3-Individual Entity Esc-Main Menu" written at the bottom of the screen.

Step 3.2

If function key F2 is pressed, then the original drawing which was drafted in AUTOCAD is displayed. Pressing function key F3 shows one entity (line,circle or arc) at a time. Pressing Esc key takes back to the main menu.

Step 4 : Inputting Blank Size

After loading the DXF file, the size of the blank should be specified. For this, the cursor is positioned over "BLANK SIZE INPUT" and carriage return is pressed. Two rectangles side by side, with some message below each of them appear on the screen. The cursor appears over "LENGTH (X)". The default value of X can be changed by pressing carriage return, typing the value of X, and again pressing carriage return. Now the default value gets changed. Similarly, the values of Y and Z can be changed. Since, the default value of Z is 0, it should be changed to some positive value.

Pressing Esc key takes back to the main menu.

Step 5 : Identifying Slots and T_slots

To identify the slots and T_slots of the part drawing, the cursor is positioned over "SLOTS & T_SLOTS IDENTIFN" and carriage return is pressed. The message "Recognizing Slots and T_slots..." appears in the screen. When the recognition of slots and T_slots is complete, the message goes off and only the main menu is left.

Step 6 : Sequencing the Operations

To sequence the different machining operations, the cursor is positioned over "OPERATIONS SEQUENCING" and carriage return is pressed. The message "Sequencing the operations..." appears on the screen. When the sequencing is over, the message goes off and only the main menu is left.

Step 7 : Generating NC codes

To generate NC codes the cursor is positioned over "NC CODES GENERATION" and carriage return is pressed. The message "Generating NC codes" should appear on the screen. If the drawing is very complicated then it will take a while. When the NC codes generation is complete the message disappears from the screen.

Step 8 : Collecting Machining Statistics

Information regarding the operations can be seen by positioning the cursor over "STATISTICS" and pressing carriage return.

Pressing Esc key takes back to the main menu.

Step 9 : Animating Machining of the Part

To see the animation of the machining of the part, the

pressed. Some message along with "Press any key to continue" is displayed in the screen.

Pressing any key shows the animation of the removal of extra material from the sides. When this is complete, "Press any key to continue" appears in the screen. Then the press of a key shows the animation of the machining of side 1. The sequence of pressing a key and animation of the machining of a side continues till the machining on all the four sides is over. Finally the main menu appears.

ERROR MESSAGES

The different Error messages and the probable reasons due to which they appear are explained below in brief.

(1) Cannot open input file.

Reasons

(i) the file does not exist in the current directory,

(ii) the file name has been typed wrongly.

(2) Error in the drawing.

Reasons

(i) the drawing is not a closed polygon,

(ii) the drawing has some inclined edges,

(iii) step 1.6 has been skipped while drafting the drawing,

(iv) the drawing contains some arcs.

(3) DXF file not loaded.

Reasons

(i) No DXF file has been loaded. A drawing with error

will be treated as if it has not been loaded.

(4) Slots & T_slots have NOT been Identified

<u>Reasons</u>

(i) Step 5 has been skipped.

(5) Sequencing has NOT been done

<u>Reasons</u>

(i) Step 6 has been skipped

REFERENCES

1. Chang, T.C. and Wysk, R.A., An Introduction to Automated Process Plannig Systems, Prentice Hall Inc. 1985.

2. AUTOCAD Reference Manual, Release 10.0, Autodesk, Inc. 1989.

3. Apte, Sanjay D., Development and Implementation of a Part Feature recognition System for Process Planning of Prismatic Parts, Unpublished M.Tech. Thesis, I.I.T., Kanpur, 1989.

4. Koren, Y. and Ben-uri J., Numerical Control of Machine Tools, Khanna Publishers, 1988.

5. Aherrao, P.K., A Generative Process Planning System for Milling Operations, Unpublished M.Tech. Thesis, I.I.T., Kanpur, 1989.

6. Kelly-Bootle, S., Mastering Turbo C, BPB Publications, 1988.

BIBLIOGRAPHY

1. Kakino, Y., et.al., A New Method of Parts Description for Computer Aided Production Planning, Advances in Computer Aided Manufacturing, North - Holland Publishers (1977), pp. 197-213.

2. Woo, T.C., Computer Aided Recognition of Volumetric Designs, Advances in Computer Aided Manufacturing, North-Holland Publishers (1977), pp. 121-135.

3. Kyprianou, L.K., shape Classification in Computer aided Design, Unpublished Ph.D. Dissertation, University of Cambridge, England (1980).

4. Choi, B.K., Barash, M.M., and Anderson, D.C., Automatic Recognition of Machined Surfaces from a 3-D Solid Model, Computer Aided Design, Vol. 16, No. 2 (march, 1984), pp. 81-86.

5. Choi, B.K., Barash, M.M., STOPP: An Approach to CAD/CAM Integration, Computer Aided Design, Vol. 17, No. 4 (May, 1985), pp. 162-168.

6. Henderson, M.R., and Anderson, D.C., Computer Recognition and Extraction of Form Features: A CAD/CAM Link, Computers in Industry (1984), pp. 324-329.

7. Raker, D., Rice, H., Inside Autocad, The Complete Autocad Guide, New Riders Publishing Micro-Tech Publications, 1989.

ABSTRACT

Name of the Student	:	Ashok Kumar Khuntia
Roll No.	:	8811401
Degree for which Submitted	:	M.Tech
Department	:	Industrial & Management Engineering Programme
Thesis Title	:	Feature Recognition, Operations Sequencing and NC Code Generation for Prismatic Parts with Linear Sweep
Name of the Supervisors	:	1. Dr. Kripa Shanker 2. Dr. S. Sadagopan
Month and Year of Submission	:	February, 1990

In the present work, a system has been developed for prismatic parts with linear sweep to recognize features, sequence the operations, generate NC codes and validate the system through animation. The prismatic part is considered to have orthogonal surfaces. It can have through holes to be drilled on top and bottom surfaces. The part can be machined by four types of cutters: end mill, slot, T_slot cutter and drill.

The input to the system is the two dimensional plan. The feature recognition system recognizes slots and T_slots through the comparision of X and Y-coordinates of consecutive points. Steps are identified by a sequencing algorithm. Steps may combine together to form virtual slots. The sequencing algorithm results in a sequence of operations, starting from the right end of the part and

finishing at the left end of the part. The NC codes, generated by the system are verified by animating the machining of the part on the computer monitor.

For the representation of lines that constitute steps, slots and T_slots and circles that indicate holes, Array data structure is used.

The drawing is drafted in AUTOCAD ver. 10.0. The system is implemented using AUTOLISP and TURBO C ver. 2.0 on an IBM compatible PC-AT having EGA monitor.

Statement of thesis preparation

1. Thesis title Feature Recognition, Operations Sequencing and NC Code Generation for Prismatic Parts with Linear Sweep

2. Degree for which submitted M. Tech

3. The thesis guide was referred to for thesis preparation Yes

4. Specification regarding thesis format have been closely followed Yes

5. The contents of the thesis were organized according to the guidelines Yes

(Signature of Student)

Name : Ashok Kumar Khuntia
Roll No. : 8811401
Department : Industrial & Management Engg. Programme